



中华人民共和国国家标准

GB/T 30881—2014/ISO/IEC 19773:2011

信息技术 元数据注册系统(MDR) 模块

Information technology—Metadata registries (MDR)—
Modules

(ISO/IEC 19773:2011, IDT)

2014-09-03 发布

2015-02-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言	I
引言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 标准结构	24
5 绑定	25
6 一致性	25
7 国际化项目的命名	25
8 概要命名	26
9 预留部分	26
10 模块 10:reference-or-literal (reflit)的数据结构	26
11 模块 11:multiple internationalized/localized values and data 的数据结构	32
12 模块 12:multiple internationalized/localized strings and texts 的数据结构	38
13 模块 13:slot tuple 的数据结构	45
14 模块 14:unstructured table of slot tuples 的数据结构	49
15 模块 15:reified relationships and relationshipssystems 的数据结构	51
16 模块 16:UPU postal data 的数据结构	53
17 模块 17:ITU-TE.164 电话号码数据的数据结构	61
18 模块 18:who-what-where-when-why-how (W5H) event data 的数据结构	65
19 模块 19:entity-person-group (EPG) 联系数据的数据结构	69
20 模块 20:entity-person-group(EPG)安全性认证数据的数据结构	72
21 模块 21:entity-person-group(EPG)关系和分组数据的数据结构	75
参考文献	79

前 言

本标准按照 GB/T 1.1—2009 给出的规则进行起草

本标准使用翻译法等同采用 ISO/IEC 19773:2011《信息技术 元数据注册系统 模块》。

本标准由中国标准化研究院提出。

本标准由全国信息技术标准化技术委员会(SAC/TC 28)归口。

本标准起草单位：中国标准化研究院、北京科迪智标信息技术有限公司、环境保护部信息中心。

本标准主要起草人：姜晓琳、孙广芝、邢立强、王晶、刘定。

引 言

本标准规定了在应用(系统)中进行使用或重复使用的数据模块。这些模块取自国家标准 GB/T 18391.3 元数据注册系统(MDR),并对其做了进一步细化。这些模块旨在协调用于规范元数据注册系统的 GB/T 18391 系列标准的当前和未来版本。

信息技术 元数据注册系统(MDR) 模块

1 范围

本标准描述了 GB/T 18391 系列标准中元数据模块的技术互操作性的细节。

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。

ISO/IEC 11404:2007 信息技术 通用目的数据类型[Information technology—General Purpose Datatypes (GPD)]

ISO/IEC 20944-1 信息技术 元数据注册系统的互操作性和绑定(MDR-IB) 第1部分:一致性框架、通用词汇及通用条款[Information technology—Metadata Registries Interoperability and Bindings (MDRIB)—Part 1: Overview, common vocabulary, and common provisions for conformance]

ISO 21090:2011 健康信息 统一数据类型信息交换(Health informatics—Harmonized data types for information interchange)

IETF RFC 2421 互联网邮件的语音配置文 版本2 (Voice Profile for Internet Mail—Version 2)

IETF RFC 3986 统一资源标识符(URI):通用语法 (Uniform Resource Identifier (URI): Generic Syntax, January 2005)

IETF RFC 3987 国际化资源标识符(IRIS)[Internationalized Resource Identifiers (IRIs), January 2005]

IETF RFC 5646 用于识别语言的标签 (Tags for Identifying Languages, September 2009)

UPU S42a-6:2009 国际邮政地址组件和模板 第1部分:概念层次和模板语言 (International postal address components and templates—Part A: Conceptual hierarchy and template languages)

3 术语和定义

下列术语和定义适用于本文件。

3.1 符号、引用及其关联

3.1.1

引用(动词) to reference

与某特定对象建立关联。

[ISO/IEC 20944-1]

示例:指某个对象的人。

3.1.2

引用(名词) reference

与特定对象的关联。

注：在某个语境中，引用与源文本不同：源文本直接给出数据，而引用指向数据，须间接进行存取、检索，或更新。

示例：相邻产生的关联；计算机内存指针；数据库外键。

3.1.3

所指对象(名词) referent

引用的结果。

[ISO/IEC 20944-1]

3.1.4

解引用 to dereference

对引用所指对象的访问。

[ISO/IEC 20944-1]

3.1.5

能指(名词) signifier

标记

通用概念，其外延是与对象相关的可感知物体。

注：通过指定的关系与一个种类“概念”的物体相结合的记号，被认为是指定概念的记号，即符号表示。

3.1.6

标记 label

通过记号给对象定位。

3.1.7

命名 designation

用指示概念的记号来表示概念。

[改编自 ISO 1087-1]

3.1.8

标识符 identifier

所指向内容的标记(label)。

注 1：标识符也是参考文献。

注 2：此定义与描述统一资源标识符(URI)句法学和语义学的 IETF RFC 2396 相一致。

3.1.9

源文本 literal

从句法的角度，源文本是代表自身的词法标记。

[ISO/IEC 2382-5]

注：在某个语境中，引用与源文本不同；引用的词法标记不代表自身，而代表其他。

示例：名称 一月、二月、三月等，下列数据类型定义为源文本：

月份_类型为(一月、二月、三月、四月、五月、六月、七月、八月、九月、十月、十一月、十二月)；

月份:月份_类型

.....

月份:=四月

代表自身的记号“一月”(即一年的第一个月)，与代表不受约束的、可以代表其他值的变量记号“一月”对比。

3.1.10

名称空间 namespace

已界定了使用范围的标签集。

[ISO/IEC 20944-1]

3.1.11

名称空间标签 namespace label

贴标签于指定的名称空间。

[ISO/IEC 20944-1]

3.2 基础数据类型

3.2.1

字符串 string

作为一个整体考虑的具有相同性质的元素(如字符或位)序列。

[ISO/IEC 2382-04:1999]

注:字符串可能为空或只包含一个元素。

3.2.2

字符串 character string

纯粹由字符组成的字符串。

[ISO/IEC 2382-04:1999]

3.2.3

文本 text

以字符、符号、字词、短语、段落、句子、表格,或其他符号排列的形式出现的数据试图传递某种意思,对它们的解释主要是基于读者对自然语言或人工语言的认识。

[ISO/IEC 2382-23:1994]

示例:一封印在纸上或显示在屏幕上的商业信件。

3.2.4

数据类型 datatype

不同的值的集合,以那些值的属性以及对那些值的操作为特征。

[ISO/IEC 11404]

3.2.5

(数据类型的)表征操作 characterizing operations (of a datatype)

用相同的值空间把一种数据类型与其他数据类型区分开的操作、收益和数据类型的值的集合。

[ISO/IEC 11404]

3.2.6

值 value

可全部用于计算的对象。

注:与一个对象对比(以面向对象计算的想法来使用),值的全体和边界是已知和确定的,而(面向对象)对象的全体或边界可以是未知或不确定的。

3.2.7

值空间 value space

一套给定数据类型的值。

[ISO/IEC 11404]

3.3 与实现相关的通用概念

3.3.1

执行时间 execution time**运行时间 run time**

某个特定程序执行发生的瞬间。

[ISO/IEC 2382-07]

3.3.2

使用语境 context of use

在使用产品、程序或服务中涉及的用户、任务、设备(硬件、软件 and 材料)以及物质条件和社会环境。

[改编自 ISO 9241-11:1998]

3.3.3

最低峰值 smallest permitted maximum; SPM

〈技术规范〉满足或超过规定,为描述某尺寸参数的最高值的实现值所指定的一个所需最低值。

[ISO/IEC 20944-1]

示例:规定“域 X 最低峰值长度应为 17”,SPM 是 17(适用于域长度)。此规定指:实施者可以执行“域 X”的最大长度为 17、18、19 等,但不能是等于或小于 16。因此,字符 X [17]满足实现要求,即使数据本身可能较小,如,一个 10 个字符的字符串存储在一个 17 个字符的数组中。

注:SPM 为由实现环境决定的最高值设定一个下限。

3.3.4

由实现环境决定(形容词) implementation-defined

〈技术规范〉未规定,但如何从备选方案中选择执行文件已经确定。

[ISO/IEC 20944-1]

例 1:由实现环境决定的特征;由实现环境决定的值;由实现环境决定的行为。

例 2:标准规定,数组 X 的大小由最小值 17 决定。此规定暗含两个要求:1) 数组的大小要大于或等于 17;2) 执行过程将证明实际大小。此例满足或超过了规定(如,最低峰值)。

注:“未规定”和“由实现环境决定”区别是,后者需要执行文件,而前者并不需要(前者也并非禁止执行文件)。

3.4 适用于多个模块的术语

3.4.1

元数据模块 metadata module

元数据注册模块

在指定的语境中,作为离散集产生、管理、使用或互换的描述型数据单元或数据组。

3.5 供将来使用的占位符

3.6 供将来使用的占位符

3.7 供将来使用的占位符

3.8 供将来使用的占位符

3.9 供将来使用的占位符

3.10 模块 10—专业术语:reference-or-literal(reflit)的数据结构

以下是对模块 10 的补充定义。

3.10.1

字符串 Character string

表示字符串的数据类型。

[ISO/IEC 11404]

注:有关的 ISO/IEC 11404 字符串数据类型采用一套合乎逻辑的字符的参数指令表。通常情况下,字符串(iso-10646-1)是用来方便地存储文本数据,即它的价值将通过数据类型的实现被保留。

3.10.2

八位字节串 **octet string**

只包含八个字节的字符串。

3.10.3

八位字节串 **octetstring**

代表八个字节的数据类型。

[ISO/IEC 11404]

注：八位字节串数据类型可以用来存储二进制数据，即其值将通过数据类型的实现被保留。

3.10.4

引用-源文本 **reflit**

数据类型，其值可以作为源文本的值直接访问或通过引用值间接访问。

3.11 模块 11—术语：**multiple internationalized/localized values and data** 的数据结构

以下是对模块 11 的补充定义。

3.11.1

多数据 **multidata**

所有具有相同的含义，但根据使用环境可能有不同表现形式和数据类型的值的集合。

3.11.2

多值 **multivalue**

所有具有相同的含义，但根据使用环境可能有不同表现形式的已定义的数据类型的值的集合。

注：多元值是定义了数据类型的多元数据。

3.12 模块 12—术语：**multiple internationalized/localized strings and texts** 的数据结构

以下是对模块 12 的补充定义。

3.12.1

文件 **document**

已命名和结构化的文本单元和图像，能够作为独立的单位被存储、编辑、检索，并在系统或使用者之间进行交换。

3.12.2

多字符串 **multistring**

所有具有相同含义，但根据使用环境可能有不同表现形式的字符串的集合。

注 1：除多字符串是专门用于字符串之外，多字符串与多值类似。

注 2：除多文本可能会以某种形式（如文件或图形），而不是用字符串来表示之外，多字符串与多文本类似。

3.12.3

多文本 **multitext**

所有具有相同含义，但根据使用环境可能有不同表现形式和数据类型的文本值的集合。

3.13 模块 13—术语：**slot tuple** 的数据结构

以下是对模块 13 的补充定义。

3.13.1

slot tuple

由标识符、种类和值组成的三元组。

3.13.2

slot_tuple

元组项的数据类型。

3.13.3

种类 kind

分类系统内的(数据分类)范围。

注 1: 通常情况下,分类系统分为外延、内涵,或两者兼而有之。在此标注中,术语“外延”(与概念对应的所有对象)和内涵(组成概念的一系列特征)在术语的领域中与它们的用法相对应。(参照 ISO 1087-1)

注 2: “数据类型”是“种类”的特化(即“种类的种类”)。以下方面数据类型不同于种类:1)数据类型暗含值空间,这是对外延的专业化(所有的值和所有的对象相对比);2)数据类型暗含以性质和操作特征为表征,这是内涵的专业化(类似数据的性质和操作特征与特点的运作与一般特点作比较);3)数据类型暗含定义了的外延和内涵,而种类既不需要外延也不需要内涵。

注 3: 术语“种类”(对象的特征)不应与种混淆(以通用的特定关系对一般概念的表示,例如,狗是一种哺乳动物)。

3.13.4

N-tuple

长度 N 的元组。

示例:元组 $\langle x, y, z \rangle$ 是一个三元元组;元组 $\langle x, y \rangle$ 是二元元组(也就是众所周知的元组对);元组 $\langle x \rangle$ 是一元元组(注 $\langle x \rangle \neq x$ 因为 $\langle x \rangle$ 是个序列,而 x 是个标量;元组 $\langle \rangle$ 是 0-type(也就是在数学中众所周知的“unit type”)。

3.13.5

元组 tuple

元素的有序列表。

示例:元组 $\langle x, y, z \rangle$ 包含三个值 x, y, z ; $\langle x, y, z \rangle$ 不等于 $\langle z, y, x \rangle$, 与集合(元素不排序)相比较,集合 $\{x, y, z\}$ 就与 $\{z, y, x\}$ 相等。

3.14 模块 14—专业术语:unstructured table of slot tuples 的数据结构

模块 14 无补充定义。

3.15 模块 15—专业术语:reified relationships and relationships systems 的数据结构

3.15.1

关系 relationship

已确定角色的零个或多个对象之间的联系。

3.15.2

关系系统 relationship system

对象及其关系的集合。

3.16 模块 16—专业术语:UPU postal data 的数据结构

以下是模块 16 中的附加定义。

3.16.1 S42a-6 的术语

3.16.1.1

收件人 addressee

邮递事件最终收件的自然人或法人。

[UPU S42a-6]

注 1: 收件人作为邮政地址的一部分可以明确定义,也可以隐含。举例来说,在某些国家,收件人信息的遗漏被视为暗示着收件人必须是具有合法进入交货点的个人或法律实体。

注 2: 术语“自然人或法人”运用在上述定义中是为了与其他标准保持一致。它应理解为包括没有法人资格的个人团体和组织形式。这也适用于对收发室管理员的定义,但不能定义邮件的发件人和邮件提出者,因为这些当事人可以从事违反邮政法规的某项活动,要承担法律上的责任。

注 3: 上述定义与 ENV 13712 中的定义不同。

注 4: 邮政运营商使用的收件人数据可能由适用于邮件的邮政服务决定。对于某些服务,如挂号邮件,邮政运营商的责任可能包括确保收件人或正式授权代表,保证收到邮件。在其他情况下,收件人数据可能仅仅是单纯的信息或所供使用的邮政运营商进行一致性检查和/或激活代理服务。在其他一些情况下,它可能被用于在投递前进行排序的目的(例如,商业邮箱会按照部门或公司负责人预先排序)。

注 5: 当收件人被显式的定义时(参见注 1),在语法正确的邮政地址中总会有一个收件人,尽管发件人信息不一定出现。在一些国家收件人可能被抽象为如“邮政客户”。

3.16.1.2

投递 **delivery**

邮政投递是指邮递项目通过移交或留给收件人、收发室管理员或授权人,或者存放在一个他们能够访问的私人信箱,而脱离邮政经营者责任范围的邮政过程。

[UPU S42a-6]

注: 在特殊的投递服务中,收件人或收发室管理员需要确认收件。除此之外,投递过程并不一定保证邮件真正到达收件人手中。尤其如果邮件放在收发室或私人信箱中,其他人可能以合法或其他手段拿到。

3.16.1.3

收件人地址 **delivery address**

由发件人指定要求邮政经营者投递邮件的邮政地址。

[UPU S42a-6]

注 1: 收信人地址可能会出现某些状况,如邮件上没有标注姓名地址。在这种情况下,收件人地址是由邮政经营者根据经营者和发件人之间的协议决定。

注 2: 邮件实际上可能没有投递到要求的收件人地址。例如,在邮件转发的情况下,收件地址为转发地址。

3.16.1.4

交货地点 **delivery point**

邮政经营者认可的一个有效的可投递邮政项目的地理位置。

[UPU S42a-6]

3.16.1.5

转发地址 **forwarding address**

由收件人或发件人指定的邮政地址,当投递地址不可达时转发的地址。

[UPU S42a-6]

注 1: 并非所有的邮件都可以被转发,对于一些邮政服务,如果邮件不能送达的话,邮件可能需要返还。

注 2: 转发地址可以是永久的,比如收件人地址,也可能是临时的。也可能将诸多收件人的信件集中保管(参见邮局代留)。

3.16.1.6

邮件收件人 **mail recipient**

实际收到邮件的个人,或是如果邮件留在收发室第一个拿到的个人。

[UPU S42-1]

注: 邮件收件人一般应该是收件人、收发室管理员或是二者之一的授权代表,但有时也并非如此。例如,假如邮件留在收发室,一个第三方能够接触到的地方;假如收件人/收发室管理员没有留下转发说明就迁出,或者假如收件人或收发室管理员说明含糊不清,导致邮政经营者曲解。

3.16.1.7

邮件提交者 **mail submitter**

负责将邮件提交到邮政系统的组织。

[UPU S42a-6]

注: 邮件提交者可能但不一定是邮件的发件人。

3.16.1.8

收发管理组织 mailee

在邮政地址中指定的组织,其有责任确保将由邮政运营商投递到送货地址的邮件送达收件人。

[UPU S42a-6]

注 1: 不同于收件人,收发管理组织是不会隐含的。如果邮政地址不包含收发管理组织的详细说明,那么就是没有收发管理组织。

注 2: 和收件人的情况一样,关于收发管理组织的说明可能会含糊不清。

注 3: 术语“自然人或法人”运用在上述定义中是为了与其他的标准保持一致。应理解为包括无法人资格的各种组织形式。这也适用于对邮件收件人的定义,但不能定义邮件的发件人和邮件提交人,因为这些当事人可能会从事违反邮政法规的行为。

3.16.1.9

邮件管理员 mailer

在邮政地址中指定的自然人或法人,其有责任确保将由邮政运营商投递到送货地址的邮件送达收件人。

[UPU S42a-6]

注: 邮递物品的生产和邮寄过程中涉及很多流程。这些流程包括:启动;内容生产,这可能是由几个不同的组织完成;整理,包括内容装配以及打包用于邮寄(例如,放置在一个信封或包裹);填写地址;汇总到邮政系统;支付。这些过程可以由一个组织完成,也可以由不同的组织分别完成,每个组织承担一个或者多个角色。当需要对这些角色进行区分时,他们被称作单个项,邮件发件人、邮件提交者和付款人;当不需要区分这些角色时,这样的区分是没有必要的,邮件管理员作为一个通用的术语。

3.16.1.10

组织 party

一个或多个自然人和(或)法人和(或)组织,没有法律的属性,作为单个实体参与邮政投递的过程。

3.16.1.11

邮政地址 postal address

使邮件能被投递到明确的实际存在或潜在的投递点的信息集,它通常包括具体的收件人和/或收发室管理员的信息。

[UPU S42a-6]

注: 邮政地址可能会含糊不清、不正确或不存在。另见语句正确的邮政地址,有效的邮政地址。

3.16.1.12

邮政地址的组成部分 postal address component

邮政地址要素、邮政地址建构与邮政地址节的集合术语。

[UPU S42a-6]

注: 3.16.15 和 3.16.16 定义了可能会发生在真实邮件地址的邮政地址组成部分。在特定的情况或某类邮政地址中,并非需要用到所有的组成部分。

3.16.1.13

邮政地址结构 postal address construct

邮政地址的元组的合并。

[UPU S42a-6]

注 1: 结构可以是层次的。

注 2: 对于特定的实例或者邮政地址类型,并非所有的结构都是需要的。

3.16.1.14

邮政地址要素 postal address element

邮政地址的基本实体。它有明确的意义和代表性,对客户或邮递处理目的具有重要意义。而且不

能以交换或印刷为目的被划分为较小单位。

[UPU S42a-6]

注1: 姓名是一个邮政地址要素,但组成姓名的单个字符则不是。另外,复姓被视为邮政地址建构,而不是邮政地址要素。因为他们需要被分为较低层次的成分——姓氏前缀和姓氏。

注2: 特定的情况或某类邮政地址中,并非需要用到所有的要素。

3.16.1.15

邮政地址元素以及元素子类型编码 postal address element and element sub-type code

邮政地址元素或元素子类型的表示,使用符合规范的简化标记,适合在模板中加以使用,相比使用名称的方法,使用编码的方法是一种语言独立的更好的方法。

[UPU S42a-6]

3.16.1.16

邮政地址元素子类型 postal address element sub-type

邮政地址元素细分,用于便利模板的设计、地址翻译、地址数据库存储以及相关技术需要。

[UPU S42a-6]

3.16.1.17

邮政地址段 postal address segment

命名与邮政地址建构和/或邮政地址要素相关的组。

[UPU S42a-6]

3.16.1.18

邮政地址结构 postal address structure

邮政地址组成部分就是邮政地址,或组合成为邮政地址的方式。

[UPU S42a-6]

注: 邮政地址结构可能因国家、地区、甚至同一国家内经营者的不同而不同。

3.16.1.19

邮政地址范本 postal address template

关于如何写邮政地址的陈述。尤其是它详细地展示了邮政地址元素出现的顺序,区分了强制性和可选性的邮政地址要素,并提供了编译说明。

[UPU S42a-6]

3.16.1.20

邮局取件 poste restant

投递服务指示表示邮件已经被投递到制定的邮政设施和代理机构。

[UPU S42a-6]

3.16.1.21

邮政数据标识符 postal data identifier

数据结构的字母前缀,其用于定义内容、格式以及数据的表示。

[UPU S42a-6]

3.16.1.22

编译说明 rendition instruction

关于当邮政地址元素被印在邮件上时,如何将它格式化,或在某些可选情况下进行格式化。

[改编自万国邮联 S42-6]

注: 编译说明反映了如何妥善地将地址格式化的规则,包括标点符号、间距、字体、邮政编码的格式、缩写、识别标志和编码位置,以及当现有的空间有限制时如何缩短和重组地址成分以确保投递成功。

3.16.1.23

回退地址 return address

当邮件不能正常的投递到投递地址或转发地址时,回退的邮政地址。

[UPU S42a-6]

注 1: “正常的投递”的解释可能依赖于对于邮件的服务特性;

注 2: 回退地址通常是发件人的邮政地址。通常不需要显式的注明,如可能是一个公司的商标、邮资标识等。有可能存在不知道回退地址的情况,这种情况下通常安装国际惯例加以处理。

3.16.1.24

语法正确的邮政地址 **syntactically correct postal address**

是指邮政地址的各个成分完全符合有关标准及有关国家或区域规则。这些标准和规则对各个成分允许的组和内部结构有限制。

[UPU S42a-6]

注 1: 示例

称呼: 先生

名: 约翰

姓: 史密斯

街道号码或地段: 4395

主道名称: 站

主道类型: 道路

城市: 潘切斯特

分投地区指示: 费尔罕

邮政编码: po16 8bq

国家: 英国

以上部分形成了一个语法正确的英国邮政地址,但如果该国是法国,那么它就不正确了,因为法国只使用数字邮政编码。

注 2: 语法正确并不意味着有效。以上不是一个有效的邮政地址,因为其投递点并不存在。

3.16.1.25

UPU

万国邮政联盟

3.16.1.26

有效的邮政地址 **valid postal address**

指其邮政地址组成部分的组合能提供一个明确单一的投递点和收件人和/或收发室管理员的邮政地址。

[UPU S42a-6]

注 1: 有效的邮政地址在语法上并不是必须要准确。如

职务: 秘书长

组织名称: CEN

街道号码或地段: 36

主道名称: 斯塔萨特

主道类型: 街

城市: 布鲁塞尔

国家: 比利时

这是一个语法不正确的邮政地址,因为邮政编码缺失。但它是有效的,因为在布鲁塞尔只有一个斯塔萨特大街(所以这个地址是明确的)。

注 2: 收件人和/或收发室管理员的详细信息可能是隐含的。就像邮件适用于有合法进入投递点的人的例子。

3.16.2 邮政地址段 **Postal address segments**

3.16.2.1

收件人规范 addressee specification

指定收件人的邮政地址段。

[UPU S42a-6]

注 1: 收件人规范由个人或组织的标示符组成,有可能包含收件人角色标识符。

注 2: 收件人规范是可选的或强制要求的,取决于邮政地址使用的邮政服务。比如,对于普通的信件,投递点规范在很多国家已经满足要求了,这种情况下收件人可以认为是在法律上可以访问投递点的个体;相反,注册的邮件通常需要附带一个显式的收件人规范。

3.16.2.2

delivery point specification 邮寄点规范

邮政地址段,其指定邮件的邮寄点。

[UPU S42a-6]

注 1: 邮寄点规范由定义的权力机构、国家、方位以及邮寄点位置和/或邮政的邮寄服务点标识符组成。邮编可能同样也是需要的。

注 2: 邮寄点规范和邮寄点之间的关系可能是服务或时间依赖。比如,地址为房间的一封普通邮件可能被邮寄到大楼大厅入口处的信箱;一个带有相同邮政地址的注册邮件必须被投递到收件人本人(或其代表),有可能直接送达该门牌号。类似的,如果客户的位置发生移动,商业回复邮件和免费邮件服务号以及邮寄点可能会发生变化。

3.16.2.3

邮件收件人调度信息 mail recipient dispatching information

当邮件收件人不是收件人(邮件地址收件人)时,为选择路径或调度邮件提供信息的邮政地址节。

[UPU S42a-6]

注 1: 邮件收件人调度信息是给收发室管理员(如果有指定的话)或邮件收件人准备的。如果收发室管理员已指定,或者收件人使用调度信息的话,则邮政经营者不使用。

注 2: 对于邮件寄给某组织,或由邮政经营者发给邮件收发室或邮局信箱的邮递物品,邮件收件人调度信息包括楼号、楼层、门牌号等信息。在更具体的服务中(如挂号邮件),这些信息组成了投递点的详述说明的一部分。可能还需要补充的调度数据。

3.16.2.4

收发管理组织规范 mailee specification

指定收发管理组织的邮政地址段。

[UPU S42a-6]

注 1: 收发管理组织由个体标识或组织标识组成,有可能与收发管理组织的角色描述符合并。

注 2: 仅当邮政运营商需要将邮件投递到个人或组织,而不是收件人时,收发管理组织规范才产生作用。

3.16.3 邮政地址结构 **Postal address constructs**

3.16.3.1

复姓 compound surname

识别家庭或提供出身标记的邮政地址结构。

[UPU S42a-6]

注 1: 复姓是个体标识的一个组成部分。它包含姓前缀和姓。

注 2: 把复姓分成两个元素是意欲用于对排序目的有意义的那部分结构,如果复姓的所有字对于排序目的都有意义,则不使用姓前缀。

注 3: 在本标准中,父姓和母姓、母亲的娘家姓等被认为是复姓。例如在特定文化中,孩子的复姓是把“儿子”或“女

儿”添加到父亲的名字(父姓)或者母亲的名字(母姓)上获得;在其他地方,人们可能会组合孩子的母姓和父亲的复姓,尽管只使用其中之一的姓氏才是人们常用的或者是法律上认可的。

注 4: 如果某个体拥有不止一个复姓,它们可能会被分开使用或组合使用。例如在特定的国家,已婚人士写姓名地址时或者用自己的复姓或用配偶的复姓,也可能两个都用。这种组合应被认为是一个复姓。

注 5: 个体拥有多个复姓,次序就是有意义的了。

3.16.3.2

国家级别的信息 country level information

邮政地址结构,其位于邮政地址之外,适用于国家或一组国家。

[UPU S42a-6]

注 1: 国家级别的信息是邮寄点标准中的一个要素。其由国家名称、国家编码、多个国家区域以及国际路由信息组成。

注 2: 作为规定,这些信息只存在于跨境邮件的地址栏中,而且在这些情况下,需要避免出现歧义。

注 3: UPU 强烈推荐国家级的信息使用大写字母加以表示,位于其他的地址元素之后,另起一行书写,同时使用邮寄国家的语言或国际通用语言进行书写。

3.16.3.3

邮寄点位置 delivery point location

标识邮寄点的邮政地址结构。邮政的经营者可能会在必要处通过人们可理解的形式来表示的地理和其他的空间坐标数据在一个邮寄点或一组邮寄点中选择一个。

[UPU S42a-6]

注 1: 邮寄点位置是邮寄点规范的组成部分。它包含主道、主道访问数据、街道号码或地段、邮寄点访问数据和扩展标识符。

注 2: 邮寄点位置与国家地区相关而且仅限于国家地区内部。

注 3: 如果投递点在地区图上的位置是很明显的,那么就不需要投递点存取数据。因此,简单的情况下,在地区图上只有一个建筑物有一个投递点,通道和街道号码或地区图都应该充足。假如是这样一种情况,在地区图上有两个住宅有多个投递点,通道和街道号或地区图和扩展名标识符多半是充足的。

3.16.3.4

个体标识 individual identification

邮政地址建构标识。以建立邮件收件人或收发管理员为目的,邮政经营者从单个个体或一组个体中选择一个。

[UPU S42a-6]

注: 个体标识符是收件人说明和收发室管理员说明的组成部分。包括地址的形式、名、复姓、姓名限定词和资格,其中各元素可能不出现、出现一次或多次。

3.16.3.5

地区 locality

识别投递点所在或附近地理区域的邮政地址建构。

[UPU S42a-6]

注 1: 地区是投递点说明的组成部分。它包括地区、近镇、城镇、行政区和投递服务限定符(限定词)。

注 2: 地区、近镇、城镇和行政区为地理上的四级定位信息。使用需求仅由要求明确识别投递点位置的地理区域级别的数目组成。因此:

——如果在一个国家有多个名称相同的城镇,地区与/或近镇将根据邮政经营者的说明来使用;

——虽然许多城镇通常被划分成地区或行政区,但是只有当城镇有多个同名的主道时,地区才需要在邮政地址中详细说明。

注 3: 移动投递点,比如活动房屋和船不会(永久地)固定在国家地区的某一位置。然而,为投递点说明的目的,才把它们与国家地区相联系。根据这种情况,它们或者符合注册地,或者符合投递点当时的位置或者将要到的位置。

3.16.3.6

组织标识 organization identification

邮政地址建构标识。以建立邮件收件人或收发室管理员为目的,邮政经营者从组织内的单个个体或一组个体中选择一个。

[UPU S42a-6]

注1:组织标识符是收件人说明和收发室管理员说明的组成部分。它包括功能、组织单位、组织名称以及法律地位。

注2:如果存在个体姓名形成个体标识的一部分这种情况,则组织标识符此个体姓名。在一个包括个体标识符和组织标识符的通讯地址中,一个识别邮件的收件人,另一个识别收发室管理员。

注3:功能和机构部门是可选择的,然后个体(组)识别作为组织的被授权的代表。如果组织名称足以清楚的识别计划中的组织,那么法律地位也可以选择。如果组织名称足以清楚的识别计划中的组织,那么法律地位也可以选择。

3.16.3.7

服务点标识符 service point identifier

邮政地址结构其标识邮政运营商的邮寄点或一组邮寄点,使用时需要引用邮政邮寄服务定义的标识符而不是其物理地址。

[UPU S42a-6]

注1:服务点标识符是邮寄点规范的要素。其由邮寄服务类型和邮寄服务标识符组成。

注2:服务点标识与国家和地区相关而且仅限于国家和地区内部。

示例:邮政信箱号、邮局取件以及商业回复邮件。

3.16.4 邮政地址元素 Postal address elements

3.16.4.1

收件人角色描述符 addressee role descriptor

在收件人说明节中标识已认可的个体或组织是收件人的邮政地址元素。

[UPU S42a-6]

注1:收件人角色描述符是当通讯地址包括多种收发室管理员说明或只有收件人说明和收发室管理员说明两种说明时,用来确保它们之间没有歧义。

注2:收件人角色描述符是可选择的。如果邮政地址包括收件人说明和收发室管理员说明两者的情况被忽略,两个节之间的区别毫无疑问可以从收件人说明、节的次序或邮政经营者和产品规则中推断。

示例:

Attn.注意

tav (ter attentie van) 注意(荷兰语)

or (indicates that two addressees are considered as alternatives)标识两个收件人为可选

and (indicates that two addressees are considered as forming a group)标识两个收件人形成一个组

3.16.4.2

备用的邮寄服务标识符 alternate delivery service identifier

邮政地址元素,其标识单个邮寄点或一组邮寄点,备用邮寄服务可以引用标识符而不是引用实际地址进行选择。

[UPU S42a-6]

注:备用邮寄服务标识符出现在邮寄点规范段中。其由备用邮寄服务类型的要素子类型以及备用邮寄服务标识符组成。一个备用邮寄服务类型是一个元素子类型,表示邮寄服务的类型。备用邮寄服务标识符是一个标识指定邮寄点的元素子类型,包含在邮寄服务类型中,包含在邮寄服务中或通过位置访问邮寄服务。

示例:私有邮箱。

3.16.4.3

建筑物/建筑 building/construction

标识建筑物/建筑类型的邮政地址元素。

[UPU S42a-6]

示例：

Bâtiment 建筑物

Block 街区

Houseboat 船屋

Mobile Home 活动房屋

3.16.4.4

国家编码 country code

邮政地址元素指定了国家、领土以及地缘政治利益区域的 ISO 3166-1 编码，邮寄点位于其中或通过其加以访问。

[UPU S42a-6]

注 1：该要素出现在邮寄点规范段中。

注 2：使用 ISO 3166-1 双字母表示。

注 3：在某些情况下，对于跨境邮件，国家编码有可能出现在地址中。

示例： FR

NL

NZ

3.16.4.5

国家名 country name

邮政地址要素指定国家、属地或地缘政治利益区域，邮寄点位于其中或通过其加以访问。

[UPU S42a-6]

注 1：该要素出现在邮寄点规范段中。

注 2：在指定的国家名中，使用的语言可能非常重要。

注 3：移动邮寄点，比如移动房屋或者船，有可能不能通过特定的国家永久访问。尽管为了满足邮寄点规范，他们也有一个关联的国家和位置。根据情况，关联的国家可能是其注册的国家，其当前所居的国家或者其打算前往的国家。

3.16.4.6

界定职权 defining authority

指定邮政经营者或其他对相关投递点说明的定义和维护负责的机构的邮政地址元素。

[UPU S42a-6]

注 1：该要素出现在邮寄点规范段中。

注 2：根据国家具体情况，投递点说明由中央政府机构，地方、市政管理机构或邮政经营者定义来定义和维护。

注 3：在具有竞争性的邮政服务环境中，投递点可能完全由某个别邮政经营者拥有或服务于个别邮政经营者。在这种情况下，投递点说明定义的权威性一般视为拥有或服务于有关投递点邮政经营者的身份地位。即使情况并非如此，不同的经营者对于指定特殊投递点也会有不同的方式。例如在英国，海斯有它自己的“DX 代码”体系，与邮局使用的邮政编码不同。

3.16.4.7

邮寄服务限定词 delivery service qualifier

指定用来做邮寄服务的办公室名称的邮政地址元素。

[UPU S42a-6]

示例：

BORDEAUX CEDEX 波尔多(企业特投)邮政编码

NANTES CEDEX 1 南特[法国西部港市](企业特投)邮政编码 1

FUTUROSCOPE CEDEX 游乐园(企业特投)邮政编码

3.16.4.8

邮寄服务指示牌 **delivery service indicator**

在通过邮寄服务类型标识的范畴中,在邮政邮寄服务所经由的位置内,指定专门投递点的邮政地址元素。

[UPU S42a-6]

示例: 邮箱号、BP (Boîte Postale)、PRIVATE BAG、邮局代留和商业回复服务。

3.16.4.9

邮寄服务限定词 **delivery service qualifier**

邮政地址元素,用来指示邮寄服务分发办公室的名称。

[UPU S42a-6]

注: 该要素出现在邮寄点规范段中。

示例: 邮箱号、BP (Boîte Postale)、PRIVATE BAG、邮局代留和商业回复服务。

3.16.4.10

地区 **district/sector**

赋予邮寄点所位于的或者所经由的城镇里或城镇附近的村庄、庄园或地区以名称的邮政地址元素。

[UPU S42a-6]

注 1: 该要素出现在邮寄点规范段中。由四个地区实例的元素子类型组成,每个实例包含类型和标识符。

注 2: 地区一般有通用的名称,其有可能是邮政或行政意义上的区域。地区可能是区域中的一个,包含一个类型和标识符结构。

3.16.4.11

门 **door**

标识位于建筑物内的投递点内或附近的公寓、场所或办公室的通讯地址元素。

[UPU S42a-6]

注: 该要素出现在邮寄点规范段以及邮件收件人分发信息段中。在每个段中,其由元素子类型门类型和门标识符组成。

3.16.4.12

扩充识别符 **extension identifier**

在国家地区内,通过投递点位置的其他组成部分,指定非唯一标识的某特定投递点的邮政地址元素。

[UPU S42a-6]

注 1: 例如,所有公寓的投递点都位于建筑物的门厅中,并以配置的信箱号或使用房间号来区别。

注 2: 如果通过投递点访问数据定义的位置或周边仅有一个投递点,则不需要扩充识别符。

注 3: 在包含多种表示形式的二级标识符的国家,在数据库表示和地址表示的形式可能不同,也可能其合并为一个通用描述符,比如扩展标识或门。

3.16.4.13

楼层 **floor**

标识投递点所位于的多层建筑中的某楼层的邮政地址元素。

[UPU S42a-6]

3.16.4.14

称呼 **form of address**

通过词(组)、简称或缩写标识个体或团体的公民身份或头衔的邮政地址元素。

[UPU S42a-6]

注 1: 该元素出现在称呼规范段以及收发管理标准段中。

注 2: 称呼包括军事、宗教、职业和敬语上的区别。

示例:

××先生
××太太
××先生和太太
××小姐
××家庭
××医生
××阁下
尊敬的××

3.16.4.15

职能 function

邮政地址元素,用于标识组织指定的角色或责任。

[UPU S42a-6]

注 1: 要素出现在收件人规范段和收发管理员规范段中。

注 2: 职能,与组织内的角色相关,应与资格相区别,资格是具体个体的内在属性。

注 3: 如果存在职能,其隐含的表示存在一个组织,尽管组织可能并不出现在地址中。

示例 1: 职能邮局局长之后可能存在一个镇和邮政编码的标识,但是可能忽略邮局的标识。

注 4: 单个收件人可能仅仅使用职能加以表示,有可能并不知道其具体的名字。

示例 2:

总经理
主任
销售经理
程序员
门卫
CEN/TC 331 秘书处

3.16.4.16

名字 given name

详述用来区别有同样复姓的并能够进入特定投递点的人名字的邮政地址元素。

[UPU S42a-6]

注 1: 此元素出现在收件人规范段以及收发管理员规范段中。在每个段内,其由元素的子类型名字第一部分、名字第二部分、名字第三部分组成。

注 2: 如果名字不止一个,名字的顺序非常重要。其中的一个有可能被定义为“首选”或者“偏好”的名字。

注 3: 名字可能被缩写(如 Ch 为 Charles 的缩写)或是只是要首字母表示。

注 4: 相对于家庭或父母标识符来说,名字与个人是相关联的。

3.16.4.17

国际路由信息 international routing information

邮政地址元素,表示如何到达一个国家、领地或地缘利益区。

[UPU S42a-6]

注: 元素出现在邮寄点规范段中。

示例: 通过 CAPE 镇送达

3.16.4.18

法律地位 legal status

标识组织法律地位的邮政地址元素。

[UPU S42a-6]

示例：

GmbH 有限责任公司（系德文 Gesellschaft Mit Beschränkter Haftung 的缩写）
 Inc. 有限公司
 Ltd. 有限责任公司
 AB 公司（系瑞典语 Aktiebolag 的缩写）
 A/S 合股公司或股份公司（系丹麦文 aktieselskab 和挪威文 Aksjeselskap 的简写）。
 OY 公司（系芬兰语 Osakeyhtiö 的缩写）

3.16.4.19

收发管理员角色描述符 **mailee role descriptor**

与个体或组织标识相结合标识已认可的个体或团体是收发管理员的邮政地址元素。

[UPU S42a-6]

注1：收发管理员角色描述符的目的是当邮政地址包括多个收发室管理员说明或只有收件人说明和收发室管理员说明两种说明时，确保它们之间没有歧义。

注2：收发管理员角色描述符是可选择的。如果邮政地址包括收件人说明和收发管理员说明的情况被忽略，两个节之间的区别毫无疑问可以从收件人说明，节的次序或邮政经营者和产品规则中推断。

示例：

c/o (care of) 转交
 p/a (per adres) 每个地址
 or (indicates that two mailees are considered as alternatives) 表示两个收发管理员可任选
 and (indicates that two mailees are considered as forming a group) 两个收发管理员可形成一个组

3.16.4.20

多国家区域 **multi-country region**

邮政地址元素，表示一个区域其中可能包含国家，领地或地缘利益区。

[UPU S42a-6]

注：元素出现在投递点规范段中。

示例：British West Indies(BWI)

3.16.4.21

姓名限定词 **name qualifier**

用来区别有相似教名或首字母的人与复姓相同的邮政地址元素。

[UPU S42a-6]

示例：

Ⅲ
 Senior
 the Third

3.16.4.22

组织名称 **organization name**

给予组织官方名称、注册商号名称或其他的正式番号的通讯地址元素。

[UPU S42a-6]

注：此元素出现在收件人规范段和收发管理员规范段中。在每个段内，其由元素的子类型公司名首部分和公司名尾部分组成。

3.16.4.23

组织单位 **organization unit**

识别组织下属部门的邮政地址元素。

[UPU S42a-6]

注：此元素出现在收件人规范段和收发管理员规范段中。在每个段内，其由元素的子类型两个组织级别组成。在

收发管理段中其进一步由组织单位的首部分和尾部分组成。

示例：

marketing department 市场部

after-sales service 售后服务

3.16.4.24

邮政编码 postcode

指定用来把邮件排序的代码的邮政地址元素。

[UPU S42a-6]

注 1：元素出现在邮寄点标准段中，由一级、二级和三级邮编组成。

注 2：在很多国家，邮政编码由两部分或更多部分构成，一部分用来识别投递排序发生的投递区域或邮政处理设备，第二部分定义该设备覆盖区域内的收发室或路线，第三部分，如果使用的话，标识专门投递点。例如，法国大多数的邮政编码用头二位数字表示“省”；英国的邮政编码分为两部分，第一部分由两个、三个或四个代码表示邮政区域；第二部分表示一个(组)收件地址。因此，此元素包含的子类型为一级邮政编码、二级邮政编码和三级的邮政编码。

注 3：邮政编码也被称为 ZIPs 或 ZIP 编码。

注 4：不是所有国家都使用邮政编码。很多情况下，它们属于补充信息，仅提供某地区的代码化表示，包括相关投递点和专用投递点的投递路线。

注 5：邮政编码与有关邮政章节的单个投递点或一组投递点相关。借由它们送达单个的收发室或者到达单个的投递路线上。尽管如此，它与其他分组参数有关，比如特殊服务。

注 6：尽管邮政编码是对国家有长期意义的，但它有专用时，(比照英国的海斯 DX 代码)可能是临时存在的。就像临时设定一个邮政编码专门处理慈善募捐呼吁的邮件。

注 7：尽管邮政编码主要为邮件排序设定，但在邮政处理流程环境之外，邮政编码也因为其他原因经常使用。特别是，许多机构在市场数据库中将潜在顾客特征与地理区域相关联时使用。

3.16.4.25

资格 qualification

标识个人在专业团体或社会中的专业或学术资历的邮政地址元素。

[UPU S42a-6]

注 1：此元素包含以下元素子类：优先资格、中间资格和继承资格(succeeding qualification)。

注 2：资格是个体的属性，应与在组织内指定角色的功能相区别。无论他/她在组织内的工作或者是职能职称发生什么变化，个体资格仍然有效。

示例：

欧盟国际工程技师

哲学博士

英国皇家学会院士

皇家学会会员

执业诉讼律师

3.16.4.26

地区 region

详细说明城镇或分布区指示符所在的国家地理或行政区的邮政地址元素。

[UPU S42a-6]

注 1：元素出现在邮寄点标准段中，由三个区域实例组成，每个实例包含一个类型和标识符。

注 2：地区一般与行政区有关，而与邮政地理无关。例如包括法国的省、德国的边境、英国的县和美国各州。参见 ISO 3166-2 各个国家的省级区划代码(国家细分代码)。

注 3：如果在国家内部，城镇和分布区指示符的合并是唯一的话，地区应该是可选的。

3.16.4.27

楼梯井 stairwell

邮政地址元素,表示在建筑物中通往楼层、门的道路。

[UPU S42a-6]

注:元素出现在邮寄点标准段以及邮件收件人分发信息段中。在每个段中,其由元素子类型和标识符组成。

3.16.4.28

街道号码或地块 street number or plot

指定投递点或投递点访问所位于的一片区域或者这片区域上的一个对象,毗邻干线的邮政地址元素。

[UPU S42a-6]

注1:这个可能会用房子或地点数字或名称的形式,并且与在地籍市政的房屋地区图登记表中定义的地区相对应。

注2:一栋建筑物/建筑跨越了几个地块,街道号码或地块可能是混合的,如6-8。这种情况下很难加以区分,我们需要使用区域或国家的知识来帮助判断。

3.16.4.29

补充投递点数据 supplementary delivery point data

提供意欲简化到投递点的通路或指示的附加数据或指令的通讯地址元素。

[UPU S42a-6]

示例:

Opposite number 23 23 的相反数

50 metres to the left of the main door 大门左侧 50 m 处

3.16.4.30

补充调度数据 supplementary dispatch data

提供额外数据或说明,目的是在邮件的投递过程中协助邮件接收人(mail recipient)的邮政地址元素。

[UPU S42a-6]

注:该元素出现在邮件收件人分发信息段中。

示例:一个内部的邮件分发码或邮站。

3.16.4.31

姓氏 surname

由用于分类的复姓的词根或一部分组成的邮政地址元素。

[UPU S42a-6]

注1:此元素出现在收件人标准段和收发管理员标准段中。在每个段中,其由子类型姓氏第一部分和姓氏第二部分组成。这可以用来检索不按姓氏排序的名称。

注2:对于那些姓出现在名之前的国家中,模板排序可以解决这种情况。

3.16.4.32

姓前缀 surname prefix

由不用于排序的复姓前缀或复姓的一部分组成的邮政地址元素。

[UPU S42a-6]

注:此元素出现在收件人标准段和收发管理员标准段中。

示例:

de

van

van de

von

3.16.4.33

通道 thoroughfare

邮政地址结构,其表示邮寄点可达的道路,道路的一部分或者路由。

[UPU S42a-6]

注 1: 元素出现在邮寄点规范段中。其子类型标识有三种形式,一级、二级和三级。对于每种形式,存在子类型表示名称、名称前缀、类型以及限定词,后两个还包含子类型——前和后。

注 2: 为了寻址的目的,通道不需要在陆地,比如水渠或河流可以认为是通道,来寻址河岸上的水上访问。

注 3: 通道名称可能唯一标识通道,也可能需要使用类型和限定词的信息加以补充。

示例 1:

San marcos
Pine Ridge
Main
6th
Charles de Gaulle

注 4: 通道名称前缀可能被用来分割相连的单词,而不必对通道名称的主要部分进行重要性排序。

示例 2:

"de la" in Avenue de la Republique
"of the" in Avenue of the Americas

注 5: 通道类型表示类别或通道的类型。当通道名称相同时,通道类型可以用来区分位置的实例。通道类型单独于通道名和通道限定词存在,因为其可能存在不同的缩写规则以及/或排序重要性,其不同于印刷出版中的相对位置。

注 6: 通道类型可能在印刷出版表示中先于或后于通道名称;其位置可能依赖于国家,区域和/或语言环境。比如, Belgium, 法语通道类型,比如, boulevard 和 drève du 通常在通道名称之前, whilst their Flemish equivalents, laan 和 dreef, 通常在通道名称之后。

示例 3:

Avenue
Beach
Canal
Lane
Place
Road
Square
Street

注 7: 通道限定词区分一个区域内的通道不同部分或实例,他们有着相同的通道名称和通道类型。

注 8: 当通道有不同的缩写规则以及/或在印刷出版表示中的位置不与通道名称或类型邻接时,通道限定词可能独立于通道名称存在。其在印刷出版中的位置:在开始处,在通道名称和通道类型之间;或者在结尾处,可能由国家,区域以及/或语言环境决定。

示例 4:

Directionals such as North, SW
Qualifiers such as Little, Upper

注 9: 二级道路标识道路或道路的一部分或其他的道路,邮寄点可以通过一级道路到达于此。

注 10: 三级道路标识道路或道路的一部分或其他的道路,邮寄点可以通过一级和二级道路到达于此。

3.16.4.34

城镇 town

标识以投递目的访问投递点所经由的投递点所在的或者附近的村庄、城镇或城市的名称的邮政地址元素。

[UPU S42a-6]

3.16.4.35

配楼 **wing**

标识投递点所位于的建筑物/建筑物的某一段,并且有时会通过它进入正门的邮政地址元素。

[UPU S42a-6]

3.16.5 邮政地址元素子类型 **Postal address element sub-types**

3.16.5.1

指示符 **indicator**

术语,用于邮政地址元素子类型的名称,表示根元素的一个逻辑部分,其与类型合并在一起构成一个标识符,其实例表示数字、字母或符号数据,在引用的一定范围内与一个元素的其他实例不同。

[UPU 42a-6]

示例:在房间 A,一个房间是一个类型,A 是一个标识符。房间 A 在一个有限的范围内应该是唯一标识符。

3.16.5.2

实例 **instance**

术语,用于邮政地址元素子类型的名称,表示根元素的一次出现。

[UPU 42a-6]

示例:在巴西,Quadra 7 是地区的一个实例的名称。

3.16.5.3

中间 **intermediate**

术语,用于邮政地址元素子类型的名称,表示一个位置,在前之后,在后之前。

[UPU 42a-6]

示例:在名称 Prof. Alex graaf van Nispen BA MKM, graaf 是中间限定词。

3.16.5.4

级别 **level**

术语,用于邮政地址元素子类型的名称,表示未指定的层次顺序。

[UPU 42a-6]

示例:组织单位级别 1。

3.16.5.5

名称 **name**

术语,用于邮政地址元素子类型的名称,表示根元素的逻辑部分,可能会使用相关元素的子类型加以区分。

[UPU 42a-6]

注:术语名称也被用于地址元素的名称,比如组织名称和名字。街道名是一个元素子类型,同时不是一个自己所属的元素,因为其属于一级、二级和三级街道,他们都是元素子类型。

示例:街道名称使用街道的内容和街道类型进一步加以区分。

3.16.5.6

部分 **part**

术语,用于邮政地址元素子类型的名称,表示根元素物理上的分支,比如单词或字符串。

[UPU 42a-6]

注 1:物理和逻辑部分使用元素子类型编码的第二位进行区分,但逻辑部分的名称使用定义在该部分的其他术语,比如类型,标识符和限定词。

注 2:Jean Claude 的名可能存储为两个物理部分,或者存储为一个根元素。如果名称是复姓,比如 Jean-Claude。

示例:对于物理部分:

名字的第二部分

姓的第二部分

3.16.5.7

位置 **position**

术语,用于邮政地址元素子类型的名称,表示根元素的一个实例,其可以与其他的实例在单个地址或地址集中加以合并,在特定的模板中加以处理。

[UPU 42a-6]

示例:补充邮寄点数据位置 1。

3.16.5.8

前 **preceding**

术语,用于邮政地址元素子类型的名称,表示中间和后之前。

[UPU 42a-6]

示例:Prof. Alex graaf van Nispen BA MKM 中, Prof.是前限定词。

3.16.5.9

前缀 **prefix**

术语,用于邮政地址元素子类型的名称,表示在另一个元素子类型前的一个位置。

[UPU 42a-6]

注:术语前缀也被用于地址元素的名称,在这种情况下表示在另一个元素前的一个位置。例如,姓的前缀在姓之前。街道名前缀是一个要素子类型,同时不是一个自己所属的元素,因为其属于一级、二级和三级街道,他们都是元素子类型。

3.16.5.10

一级 **primary**

术语,用于邮政地址元素子类型的名称,表示一个高于二级和三级的状态或级别。

[UPU 42a-6]

注:一级、二级和三级既可以用于实例,也可以用于部分。

示例:一级街道

一级邮政编码

3.16.5.11

限定词 **qualifier**

术语,用于邮政地址元素子类型的名称,表示根元素的逻辑部分,根据相关元素子类型进一步加以区分。

[UPU 42a-6]

注:术语前缀也被用于地址元素的名称,在这种情况下由相关元素的内容加以区分。例如,名称限定符进一步使用名和姓加以区分。街道限定符是一个要素子类型,同时不是一个自己所属的元素,因为其属于一级、二级和三级街道,他们都是元素子类型。

示例:街道名称使用街道的内容和街道类型进一步加以区分。

3.16.5.12

二级 **secondary**

术语,用于邮政地址元素子类型的名称,表示一个低于一级和高于三级的状态或级别。

[UPU 42a-6]

注:一级、二级和三级既可以用于实例,也可以用于部分。

示例:二级街道

二级邮政编码

3.16.5.13

后 succeeding

术语,用于邮政地址元素子类型的名称,表示前和中间之后。

[UPU 42a-6]

示例:Prof. Alex graaf van Nispen BA MKM 中, MKM 是后限定词。

3.16.5.14

三级 tertiary

术语,用于邮政地址元素子类型的名称,表示一个低于一级和二级的状态或级别。

[UPU 42a-6]

注:一级、二级和三级既可以用于实例,也可以用于部分。

示例:三级街道

三级邮政编码

3.16.5.15

类型 type

术语,用于邮政地址元素子类型的名称,表示根元素的一个逻辑部分,其可能与指示符合并构成一个标识符,其实例描述一个类别。

[UPU 42a-6]

示例:RESIDENCE MASUREL 中 RESIDENCE 是一个类型。

3.16.6 其他术语和定义

3.16.6.1

未提供的地址 unrendered address

与格式设置无关的邮政地址。

示例:根据 UPU S42-1 作为一系列邮政地址节表现的邮政地址为 unrendered address。

3.17 模块 17—专业术语:ITU-TE.164 电话号码数据的数据结构

下列是模块 17 的附加定义。

3.17.1

电话分机 extension

交换机上的电话号码。

示例:

0012125551212 (符合一些欧洲国家的编号计划的电话号码)

0-22-749-0111 (瑞士本地电话号码)

3.17.2

国际编号计划 international numbering plan

符合 ITU-T 的 e.164 的电话号码。

示例:

+12125551212

+41-22-749-0111 (省略标点符号和空格)

3.17.3

电话号码元素 phone number element

电话号码部分。

示例：

电话号码+12125551212 +234 可以分解为下列元素：+（国际编号计划前缀）、1（国家代码）、212（电话编号区域）、555（中心局代码）、1212（线路）、234（分机号）。

电话号码+41227490111 可以分解为下列元素：+（国际编号计划前缀）、41（国家代码）、22（城市代码）、7490111（当地号码）。

3.17.4

专用交换机 PBX private branch exchange

专用的电话设备和服务。

注：通常情况下，每一台交换机上的电话都有一些自己的号码-这些号码叫作分机号。

3.17.5

专用编号计划 private numbering plan

可以不符合 ITU - TE.164 的电话号码。

注：短语“可以不符合”指“允许不符合”。

示例：

0012125551212（符合一些欧洲国家编号计划的电话号码）

0-22-749-0111（瑞士本地的电话号码）

3.18 模块 18—专业术语：**who-what-where-when-why-how (W5H) event data** 的数据结构

无模块 18 的附加定义。

3.19 模块 19—专业术语：**entity-person-group (EPG) 联系数据**的数据结构

无模块 19 的附加定义。

3.20 模块 20—专业术语：**entity-person-group(EPG) 安全性认证数据**的数据结构

下列为模块 20 的附加定义。

3.20.1

认证 authentication

对实体所声称身份的验证行为。

[ISO/IEC 2382-08:1998]

3.20.2

访问控制 access control

确保只有接受了委托的实体才能访问数据处理系统资源的方法。

[ISO/IEC 2382-08:1998]

3.21 模块 21—专业术语：**entity-person-group(EPG) 关系和分组数据**的数据结构

3.21.1

entity-person-group

EPG

实体和/或人的实体、人或团体。

4 标准结构

本标准由 3 部分组成：

- 3.10~3.21:模块的专用术语。
- 第 10 章~第 21 章:模块的具体规定,如数据结构的定义、数据类型和其他互操作性要求。
- 其他:对所有元数据模块均适用。

每个关于模块的章节结构(第 10 章~第 21 章)由以下组成:

- 第 x.1 条 模块的介绍:对相关章、章目的,及成因的技术内容的具体信息或评论;
- 第 x.2 条 模块的范围:模块范围的描述,类似于标准的范围章节;
- 第 x.3 条 功能能力:模块的预期能力和应用;
- 第 x.4 条 抽象模型:表现为使用 UML 对象建模的数据结构的抽象概念,包括类、属性和关系的描述。虽然使用了类模块,但是类依然被认为是“简单的、旧的数据结构”,而且无需面向对象实现;
- 第 x.5 条 计算描述和数据类型:把对象模型绘制出来,并以数据类型,组件和子组件呈现。ISO/IEC 11404 中使用了数据类型符号,其为独立绑定。而术语“类”和“数据类型”指实施特异性的不同程度,每个类都与它对应的数据类型享有相同的概念;
- 第 x.6 条 绑定的附加条款:除第 5 条以外,某些模块可能有绑定的附加条款;
- 第 x.7 条 一致性的附加条款:除第 6 条以外,某些模块可能会有遵从、一致性和一致性标签的附加规定。

下列规定适用于每个模块章、条(第 10 章及以后):

- 数据类型用 ISO/IEC 11404 的标记法进行定义;
- ISO/IEC 11404 注释中的条款为规范性条款;
- ISO/IEC 11404 注释中的条款“所有组件可选”的意思是(规范的),必选属性对所有包含在结构之内的数据元素是“可选的”;
- 缩写 SPM 表示“最小的最高允许值”。SPM 值是在一致性实施时给执行者一个下限。除非已有安排,否则应用程序不应假设实施支持能力超出 SPM 值;
- ISO/IEC 11404 注释中的条款“所有大小均为 SPM”表示(规范的)大小值是最小的最高允许值。

5 绑定

应符合 ISO/IEC 20944 互操作性绑定。

6 一致性

参考文献中包含 ISO/IEC 20944-1 的第 4 章——一致性。ISO/IEC 19773 的所有章、条都应符合一致性要求。

7 国际化项目的命名

7.1 后缀句法的命名

国际化项目的命名在 ISO/IEC 指南第 2 部分的附录中有详细说明。

标准命名模式 ISO/IEC 19773 带有后缀/c/×××××相当于“ISO/IEC 19773,第×××××条”。使用连字符(-)来指定章节的范围,例如,“ISO/IEC 19773/c/10-14”是指第 10 条~第 14 条。使用逗号(,)来指定非连续的值或范围,例如,“ISO/IEC 19773/c/10-12,14”指第 10 条~第 14 条,但不包括第 13 条。

示例:产品或服务标记着“符合 ISO/IEC 19773/c/19”,意指产品或服务符合 EPG Contact Data 第 19 条(暗含也符合其他章)。

7.2 概要的后缀命名

命名模式 ISO/IEC 19773 带后缀/p/×××××相当于在第 8 条中对概要命名所定义的“ISO/IEC 19773,概要×××××”。

8 概要命名

以下是国际标准的标准概要名称。

——“16-21090”:模块 16,邮政数据对应于 ISO 21090 概要,例如在 ISO/IEC 19773 模块 16 的实现中使用 ISO 21090 的实现(参见 16.7)。

9 预留部分

预留部分。

10 模块 10: reference-or-literal (reflit)的数据结构

10.1 模块介绍

模块 10 介绍了用于存储源数据(即原始数据值)或源数据引用(例如,指向数据的指针)的引用/源数据结构。相比编译阶段选择的方法(如在编写程序或查询时,选择使用数据或数据的引用)和制定标准阶段选择的方法[比如在技术规范中(如标准)加以选择],本描述的方法在运行阶段对源数据或源数据引用加以选择。

10.2 模块范围

本章对运行阶段需要选择的源数据或源数据引用的数据存储结构进行描述。

10.3 模块功能

reflit 的数据结构允许数据单元(值)(通过源数据)直接表示,或(通过源数据的引用)间接表示,并且供运行阶段对源数据或源数据引用加以选择。

示例 1:通过使用 reflit 数据结构,描述信息可直接(如字符串源文本)或间接(如指向文档的 URL)进行存储。以下是使用字符串作为基本数据类型的例子:

```
description: reflit(characterstring) // a reference to or literal of a characterstring
```

下列代码把字符串"a large ball"作为(字符串)源文本来储存,即,描述信息包含了字符串的副本:

```
description.reflit_kind = literal;  
description.literal_value_as_text = "a large ball";
```

下列代码把字符串"a large ball"作为引用(到字符串)来储存,即,描述信息包括对含字符串的文档的指针:

```
description.reflit_kind = reference;
description.reference_value_as_text = "http://website.com/ball_description.txt";
```

reflit 数据结构提供了在运行阶段加以选择的能力,无论值可以直接获取还是不能直接获取。当对文本数据或引用数据的选择无法在使用前得知,或预先不知在何处作出选择时,会导致无法实现的执行、使用、存储或数据,此时,采用在运行阶段加以选择的方法是有利的。

示例 2: 在数据库中储存少量的描述信息(如一个段落)比较实际,但在数据库中存储大量的描述信息(例如、一部书或一部电影)就可能不切实际。通过利用 reflit 数据结构,小的字符串可以会直接存储,而大字符串会采用引用加以存储。

示例 3: 不管描述信息的大小,都应当有单个描述信息的副本,以便当描述信息发生变化时,唯一的字符串也随之变化。即整个字符串从"a large ball"到"a large red ball"的变化受文档内容"ball_description.txt"的变化的影响。在这种情况下,可以使用引用文件。

```
description_1.reflit_kind = reference;
description_1.reference_value_as_text = "http://website.com/ball_description.txt";
description_2.reflit_kind = reference;
description_2.reference_value_as_text = "http://website.com/ball_description.txt";
```

示例 4: 无论描述数据大小,每个描述数据都应当有自己的副本,以便它们能够独立地进行更新。在这种情况下,应使用源数据。

10.4 抽象模型

10.4.1 概述

概念上,reflit 是指向数据类型的指针(引用)与数据类型的实例(源数据)的联合体。我们需要一些额外数据来标识引用或源数据值的类型、编码方式(关注数据表示、而非数据类型),以及文本流的表示形式(字符数组)或二进制流(八位字节数组)的表示形式。此节剩余部分对 reflit 对象-模型进行描述。此对象模型对应 10.6 中的绑定-独立语法(binding-independent semantics)部分。

10.4.2 reflit(of_type)

reflit 是一个类,用于实例化生成“引用或源数据”。类需要 of_type 作为实例化的一个参数。示例:“reflit(characterstring)”数据结构用于存储“characterstring”数据类型(如“hello world”),或指向字符串的一个指针(如“http://hello.com/world.txt”);“reflit(space_extent)”数据结构用于存储“space_extent”数据类型(如“30N35W-40N45W”)或指向时空区域的一个指针(& lat_lon_range)。此类包括以下部分与属性:

reflit_kind: state(reference, literal, none),reflit_kind 可能取如下值,reference (表示 reflit 保存一个引用)、literal (表示 reflit 保存一个源数据)以及 none (表示 reflit 不保存数据,如,空 reflit 对象)。

reference_value: reference_type(of_type),引用,在对数据采用非直接访问方法时,对该项解引用以获取实际的数据值。

literal_value: literal_type(of_type),源数据,reflit、reference_type 及 literal_type 的 Uml 表示见图 1。

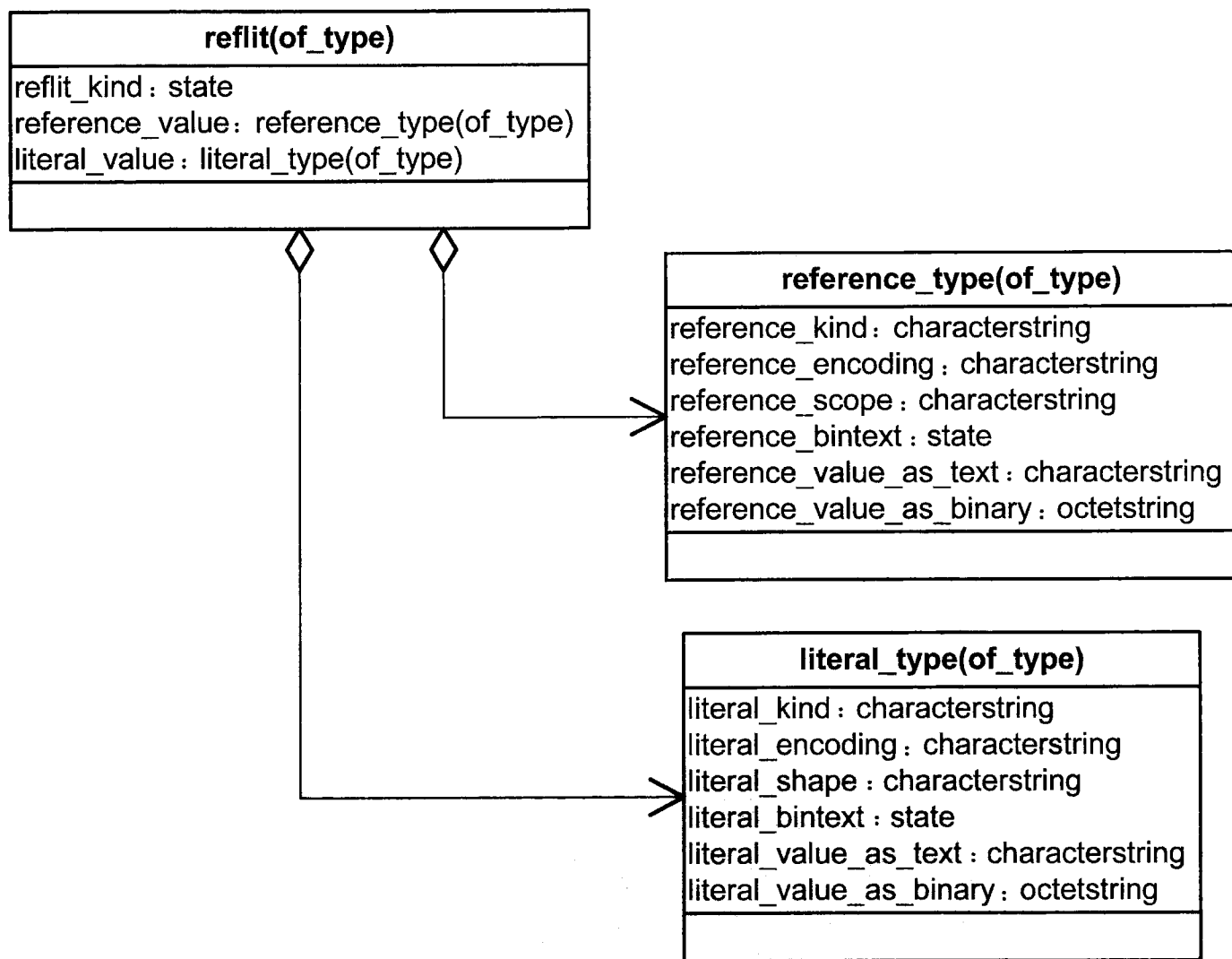


图 1 UML 表示 reflit、reference_type、literal_type

示例：下列是上面 UML 图表中可操纵对象的例子：

```

self.reflit_kind
self.reference_value
self.reference_value.reference_kind
self.reference_value.reference_encoding
self.reference_value.reference_scope
self.reference_value.reference_bintext
self.reference_value.reference_value_as_text
self.reference_value.reference_value_as_binary
self.literal_value
self.literal_value.literal_kind
self.literal_value.literal_encoding
self.literal_value.literal_bintext
self.literal_value.literal_shape
self.literal_value.literal_value_as_text
self.literal_value.literal_value_as_binary
    
```

10.4.3 reference_type(of_type)

reference_type 类保存指向数据的引用。类使用 of_type 作为实例化的参数。类包括下列部分与属性：

reference_kind: characterstring, 引用的类型, 如“uri”。reference_kind 属性描述了引用指向的数据功能属性, 而非引用本身表示的属性。此数据元有下列值：

——“uri”：统一资源标识符, 参照 RFC 2396。

- "pointer": 国际化资源标识符,参考 ISO/IEC 9899。
- "filename": 文件名(路径名),参考 ISO/IEC 9945。
- "other": 国际标准外的其他格式的引用。
- "x- *": 实现定义格式,遵循 IANA 第十章(修订)。

reference_encoding: characterstring,表示方法,如,"far32/uint64"(一个 far-32 作为一个无符号 64 位整数),"near16/hex"(near-16 指针作为一个十六进制数)、"ascii"(ASCII 文本编码)。后缀"LE"表示 little-endian 字节序,而"BE"表示 big-endian 字节序;无后缀表示字节序未指定。此数据元允许有下列值:

- ("near", "_", ("8" | "16" | "40" | "48" | "56" | "64" | "72" | "96" | "128" | "256")): 近指针,即简单地址空间的指针。"near"后的数字是在指针中的精度的位数,如"near_32"是简单地址空间的 32 位指针。
- ("far", { "_", ("8" | "16" | "32" | "40" | "48" | "64") }): 远指针,即,分段地址空间的指针。在"far"之后的数字是作为一个平面地址空间的上节最后一部分指针上节点的数目,如"far_16_32"是一个 48 位的指针,表示 32 位简单地址空间的所有 2^{16} 个可能的段。
- ("pic", "_", ("8" | "12" | "16" | "24" | "32" | "40" | "48" | "64")): 位置无关编码 (PIC) 指针。其是一个相对于使用位置的指针。在 "PIC" 之后的数字表示指针的精度位,如, "pic_16" 是一个 16 位的相对指针。
- ("uint", "_", ("8" | "16" | "24" | "32" | "48" | "64" | "128" | "256" | "512" | "1024")): 无符号整数补码。"uint"是整数中精度位数,如"uint_16"是一个 16 位无符号整数,范围区间是 $[0, 2^{16} \sim 1(65535)]$ 。
- ("int", "_", ("8" | "16" | "24" | "32" | "48" | "64" | "128" | "256" | "512" | "1024")): 有符号整数补码。"int"后的数字表示整数精度,如,"int_16"是一个 16 位有符号整数,范围区间是 $[-2^{15}(-32768), 2^{15} \sim 1(+32767)]$ 。
- ("tcdi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }): 无符号的十进制整数的正文程序代码由一或多数字从 0~9 组成,表示为 ISO 从 0~9 的 646 个字符。在"tcdi"之后的数字是在整数的精确字节数,如,"tcdi_4"是包括从 0000~9999 的 4 位整数。
- ("bcdi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }): 用二进制的值 0000~1001 表示十进制数字 0~9 的二进制编码的十进制整数。在"bcdi"之后的数字是在整数的精确字节数,如,"bcdi_4"是包括从 0000~9999 的 4 位整数。
- ("tchi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }): 使用 ISO 从 0~9 和从"A"~"F"表示十六进制数字的 646 个字符的十六进制整数的正文程序代码。在"tchi"之后的数字是在十六进制整数的精确字节数,如,"tchi_4"是包括从 0000~FFFF 的 4 位整数。
- "pointer": 实现定义的指针,参考 ISO/IEC 9899。
- "filename": 文件名,参考 ISO/IEC 9945。
- "other": 引用文件指定以一种国际标准以外的格式进行描述。
- "x- *": 实现定义的格式。

reference_scope: characterstring,理解该引用的范围(例如,边界)。如,"proc/1234"(进程 1234 的地址空间内的指针)、"stackframe/0x27463720"(在程序设计语言中的范围链接结构)、"namespace.org/object_id_list"(命名空间标签,提供命名空间的标识符)。“namespace.org/object_id_list”结构与格式的描述不在本部分中叙述。

- ("stackframe/", hex_digit_string): 引用位于执行栈中,由 hex_digit_string 指定;
- ("proc/", characterstring): 引用在进程地址空间中 characterstring 表示进程标识符 i;

- ("xmlDOM/", characterstring): 引用位于 XML 文档对象模型框架中,由 characterstring 加以指定;
- ("other/", characterstring): 引用位于 characterstring 定义的范围内,其是一个实现定义的范围;
- reference_bintext:state(binary, text, none),可能取以下值,binary(表示 reference_value_as_binary 保存引用值)、text(表示 reference_value_as_text 保存引用值),none(表示没有值,如,空的 reference_type 对象);
- reference_value_as_text:characterstring,引用值的字符串表示;
- reference_value_as_binary:octetstring:引用值的八进制表示。

10.4.4 Literal_type(of_type)

literal_type 类存储源数据值。类使用 of_type 作为实例化的一个参数。这个类别包括下列成员和属性:

literal_kind:characterstring,源数据类型,即数据类型。属性 literal_kind 描述了源数据值的功能属性,而非表示的属性。在格式“datatype // datatype_source”中的字符串,如“int // Java”对应 Java 编程语言中的 int 数据类型。

literal_encoding:characterstring,表示方法,如,"ascii"(ASCII 编码)、"utf8"(ISO/IEC 10646 UTF-8 编码)。该项允许有下列值:

- ("near", "_", ("8" | "16" | "40" | "48" | "56" | "64" | "72" | "96" | "128" | "256")):近指针,即,简单地址空间的指针。“near”后的数字是在指针中的精度的位数,如,“near_32”是简单地址空间的 32 位指针。
- ("far", { "_", ("8" | "16" | "32" | "40" | "48" | "64") }):远指针,即,分段地址空间的指针。在“far”之后的数字是作为一个平面地址空间的上节最后一部分指针上节点的数目,例如"far_16_32"是一个 48 位的指针,表示 32 位简单地址空间的所有 2^{16} 个可能的段。
- ("pic", "_", ("8" | "12" | "16" | "24" | "32" | "40" | "48" | "64")):位置无关编码(PIC)指针。其是一个相对于使用位置的指针。在“PIC”之后的数字表示指针的精度位,如,"pic_16"是一个 16 位的相对指针。
- ("uint", "_", ("8" | "16" | "24" | "32" | "48" | "64" | "128" | "256" | "512" | "1024")):无符号整数补码。“uint”是整数中精度位数,如“uint_16”是一个 16 位无符号整数,范围区间是 $[0, 2^{16} \sim 1(65535)]$ 。
- ("int", "_", ("8" | "16" | "24" | "32" | "48" | "64" | "128" | "256" | "512" | "1024")):、有符号整数补码。“int”后的数字表示整数精度,如,“int_16”是一个 16 位有符号整数,范围区间是 $[-2^{15}(-32768), 2^{15} \sim 1(+32767)]$ 。
- ("tcdi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }):无符号的十进制整数的正文程序代码由一或多数字从 0~9 组成,表示为 ISO 从 0~9 的 646 个字符。在“PIC”之后的数字是在整数的精确字节数,如,“tcdi_4”是包括从 0000~9999 的 4 位整数。
- ("bcdi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }):用二进制的值 0000~1001 表示十进制数字 0~9 的二进制编码的十进制整数。在“bcdi”之后的数字是在整数的精确字节数,如,“bcdi_4”是包括从 0000~9999 的 4 位整数。
- ("tchi", "_", { ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9") }):使用 ISO 从 0~9 和从“A”~“F”表示十六进制数字的 646 个字符的十六进制整数的正文程序代码。在“tchi”之后的数字是在十六进制整数的精确字节数,如,“tchi_4”是包括从 0000~FFFF 的 4 位整数。

- ("le" | "be" | "un"), "_", ("0" | "1" | "4" | "8" | "16" | "24" | "32" | "48" | "64" | "96" | "128" | "256" | "512" | "1204"): 在 little endian (LE), big endian (BE), or unspecified (UN) 字节序下定义的位数大小的八进制表示,由下划线后的数字加以指定。
- "pointer": 实现定义的指针,参考 ISO/IEC 9899。
- "filename": 文件名,参考 ISO/IEC 9945。
- "other": 引用文件指定以一种国际标准以外的格式进行描述。
- "x- * ": 实现定义的格式。
- literal_shape:characterstring:运行时源数据的大小,如,一个 characterstring 数据类型在初始时没有固有的大小,在实例化是为其分配不同的大小。此字符串有三种情况:1)空串(" ")表示标量的形状,2)" integer_value ",十进制数字,表示数组的长度,例如,"7"表示长度为7的数组 3)"(integer1 integer2... integerN),由空格分隔的十进制数值串。例如"(723)"表示一个 $7 \times 2 \times 3$ 的多维数组。
- reference_bintext:state(binary, text, none):可能取以下值:binary(表示 reference_value_as_binary 保存引用值)、text(表示 reference_value_as_text 保存引用值),none(表示没有值,例如,空的 reference_type 对象)。
- reference_value_as_text:characterstring:引用值的字符串表示,如"hello world"。
- reference_value_as_binary:octetstring:引用值的八进制表示,如{ 68 65 6C 6C 6F 20 77 6F 72 6C 64}。

10.5 计算的数据描述和数据类型

10.5.1 概述

reflit 数据类型是基本数据类型,不依赖任何其他模块。

10.5.2 引用-类型

在 ISO/IEC 11404 中的定义:

```
type reference_type(of_type) = record
```

```
(
```

```
    reference_kind:
```

```
        characterstring, // SPM: 8192 characters
```

```
    reference_encoding:
```

```
        characterstring, // SPM: 8192 characters
```

```
    reference_scope:
```

```
        characterstring, // SPM: 8192 characters
```

```
    reference_bintext:
```

```
        state(binary, text, none),
```

```
    reference_value_as_text:
```

```
        characterstring, // Shall be large enough to hold "pointer(of_type)".
```

```
    reference_value_as_binary:
```

```
        octetstring, // Shall be large enough to hold "pointer(of_type)".
```

```
),
```

描述:

参见 10.4.3。

10.5.3 源数据-类型

ISO/IEC 11404 定义:

```
type literal_type(of_type) = record
(
    literal_kind:
        characterstring, // SPM: 8192 characters
    literal_encoding:
        characterstring, // SPM: 8192 characters
    literal_shape:
        characterstring, // SPM: 8192 characters
    literal_bintext:
        state(binary, text, none),
    literal_value_as_text:
        characterstring, // Shall be large enough to hold "of_type".
    literal_value_as_binary:
        octetstring, // Shall be large enough to hold "of_type".
),
```

描述:

参见 10.4.3 的内容。

10.6 绑定附加条款

引用文件的范围(即其指向的对象类型)是实现定义的。

引用文件的可见度(即其在哪里被使用)是实现定义的。

引用文件的生命周期(即引用文件有效的范围)是实现定义的。

10.7 一致性附加条款

无一致性附加条款。

11 模块 11: multiple internationalized/localized values and data 的数据结构

第 11 章描述了多值,即一组具有不同的表示方式,但有同样的意义的值。多值可能被用于需要多种表现形式的应用程序中,如概念,而表示方式基于使用的语境会有所不同。这些差异可能包括国家、区域、文化、语言、交流方式(如视觉)、交流方法(如手语)、人种、环境等。除了多元数据对于每种不同的表示可以有不同的数据类型,而多值有相同的基本数据类型外,多元数据和多值非常类似。

11.1 模块简介

11.2 模块范围

本章描述了用于保存具有相同含义但有不同表示方式的值的数据结构。

注:表现形式可能由于下面的差异而不同,祖国、区域、文化、语言、交流方式(如视觉)、交流方法(例如手语)、人种、环境、等。

11.3 模块功能

11.3.1 概述

11.3 描述这章的预定能力和应用。

11.3.2 多值数据结构

multivalued 数据结构存储同一概念的多个不同的表现形式,每种表现形式与特定的语境相关。

注: multivalued 数据结构与在第 12 章中定义的 multistring 数据结构相似,但有以下差异: multistring 定义了基本的数据类型 characterstring,而 multivalued 通过用户自定义基本数据类型参数化。

示例:数据类型 dance_performance_venue 定义为:

```
type dance_performance_venue = record
(
location: characterstring, // location of event
timing: characterstring, // timing of event
),
```

```
example_instance: dance_performance_venue = ( "New York", "2000-01-01" ),
```

基于数据类型 dance_performance_venue 的多值可用于储存英式英语、美式西班牙语的表现形式:

```
type multivalued_dance_performance_venue = multivalued(dance_performance_venue), // creates the new datatype
```

```
event_in_new_york_city: multivalued_dance_performance_venue = // instantiates the new datatype
```

```
(
contextualized_value_list =
(
////////////////////////////////////
( // British English presentation
context_identifier = 'language="en-GB"',
value = ( where = "New York", when = "17.03.2002" ),
),
////////////////////////////////////
( // US Spanish presentation
context_identifier = 'language="es-US"',
value = ( where = "Nueva York", when = "03/17/2002" ),
),
////////////////////////////////////
),
),
```

11.3.3 多元数据的数据结构

multidata 数据结构存储着相同概念的不同表现形式,每一种表现形式与特定的语境相关,同时,允许对于不同的数据类型有各自的表示方法。

注: multidata 数据结构类似于 multivalued 数据结构,但有以下差异: multivalued 在每个 multivalued 数据结构的实例中有相同的用户定义基本数据类型。而 multidata 有一个用户定义的数据类型,对于不同的表现形式可能不同。

示例:下面的例子描述了用于存储概念“the number 2”多种表示形式的 multidata 数据结构:

```
concept_of_number_2: multidata = // instantiates the multidata data structure
(
contextualized_value_list =
(
////////////////////////////////////
( // English text presentation (literal value stored)
context_identifier = 'language="en", mime-content-type=text/plain',
data = ( reflit_kind = literal, literal_value =
(
literal_kind = "text",
literal_encoding = "ascii",
literal_bintext = text,
literal_value_as_text = "two",
) ),
),
////////////////////////////////////
( // German HTML text presentation (reference to URL)
context_identifier = 'language="de", mime-content-type=text/html',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
reference_value_as_text = "http://zwei.org/zwei.html",
) ),
),
////////////////////////////////////
( // graphical presentation of numeral 2 (reference to GIF image)
context_identifier = 'mime-content-type=image/gif',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
reference_value_as_text = "http://zwei.org/2.gif",
) ),
),
////////////////////////////////////
( // German audio presentation of numeral 2 (reference to WAV file)
context_identifier = 'mime-content-type=audio/wav',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
```

```

        reference_value_as_text = "http://zwei.org/zwei.wav",
    ) ),
),
////////////////////////////////////
( // Morse Code presentation of numeral 2 (literal value stored)
  context_identifier = 'mime-content-type=other/x-morse-code',
  data = ( reflit_kind = literal, literal_value =
    (
      literal_kind = "text",
      literal_encoding = "ascii",
      literal_bintext = text,
      literal_value_as_text = "..---",
    ) ),
),
////////////////////////////////////
),
),

```

11.4 抽象模型

11.4.1 概述

11.4 是 multivalue 数据结构以及 multidata 数据结构的对象模型描述。对象模型对应于 11.6 绑定独立的语法。

11.4.2 多值

11.4.2.1 概述

图 2 是 multivalue 和 contextualized_value 类的 UML 表示形式。

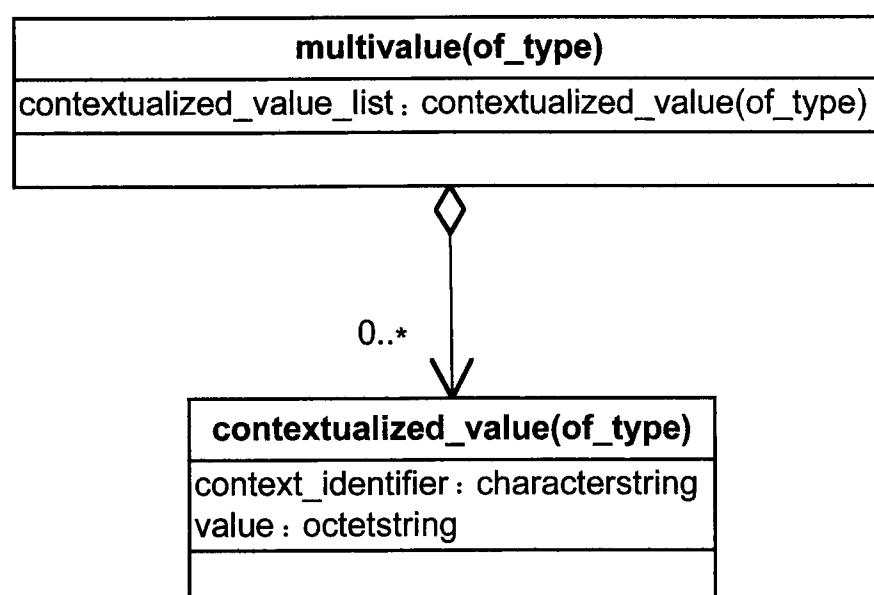


图 2 Multivalue 和 contextualized_value 的 UML 表示

11.4.2.2 多值

multivalue 类包含了下列的成员和属性：

——contextualized_value_list: contextualized_value(of_type)的 array (0..*):这个值的数组有相同的含义,每一个都有自己的语境。

11.4.2.3 contextualized_value

contextualized_value 类包含如下的成员和属性:

——context_identifier: characterstring:与语境相关的标识符,如对语境的描述。

——value: of_type:of_type 数据类型的值的表示。

示例:上述 UML 图中访问数据对象的方法例子:

```
self.contextualized_value_list
self.contextualized_value_list[0].context_identifier
self.contextualized_value_list[0].value
self.contextualized_value_list[1].context_identifier
self.contextualized_value_list[1].value
```

11.4.3 multidata

11.4.3.1 概述

图 3 是 multidata 和 contextualized_data 类的 UML 表示形式。

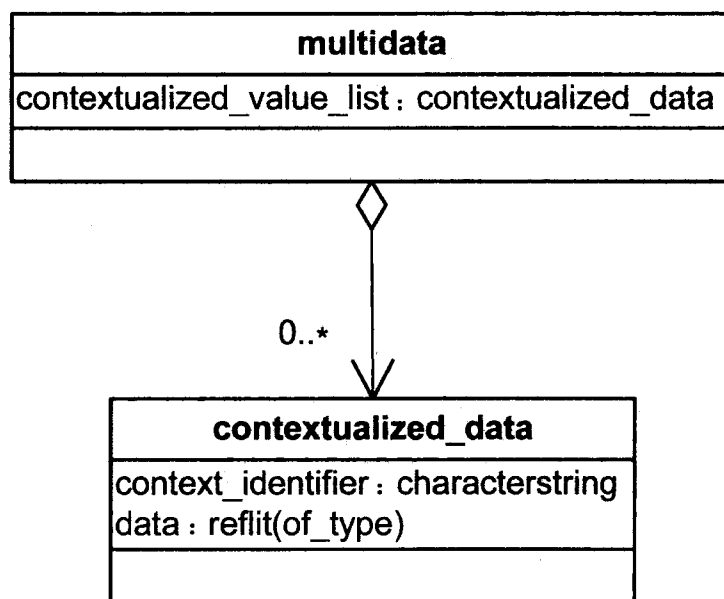


图 3 Multidata, contextualized_dara 的 UML 表示

11.4.3.2 multidata

这个 multidata 类包括如下的成员与属性:

——contextualized_value_list: contextualized_data 的 array (0..*):这个值的数组有相同的含义,每一个都有自己的语境。

11.4.3.3 contextualized_data

contextualized_data 类包含如下的成员与属性:

——context_identifier: characterstring:与语境相关的标识符,如对语境的描述。——data: reflit (of_type):同语境标识符相关联的专用表现形式独立的表现形式。reflit 数据结构在第 10 章中作出了定义。

示例:上述 UML 图中访问数据对象的方法例子:

```
self.contextualized_value_list
```



```

self.contextualized_value_list[0].context_identifier
self.contextualized_value_list[0].data // See footnote F1F
self.contextualized_value_list[1].context_identifier
self.contextualized_value_list[1].data

```

11.5 计算描述与数据类型

11.5.1 概述

第 10 章的 reflit 数据类型将在下列定义中使用。

11.5.2 multivalued

11.5.2.1 概述

11.5.2 定义了 multivalued 数据结构的数据类型。

11.5.2.2 多值

ISO/IEC 11404 定义

```

type multivalued(of_type) = record
(
    contextualized_value_list:
        array (0..*) of contextualized_value_type(of_type),
),

```

描述

参见 11.4.2.2 中对 record 和其成员属性的数据描述。

11.5.2.3 contextualized_value_type(of_type)

ISO/IEC 11404 定义

```

type contextualized_value_type(of_type) = record
(
    context_identifier:
        characterstring, // SPM: 4096 characters
    value:
        of_type,
),

```

描述

参见 11.4.2.3 中对 record 和其成员属性的数据描述。

11.5.3 multidata

11.5.3.1 概述

本条定义了用于 multidata 数据结构的数据类型。

11.5.3.2 multidata

ISO/IEC 11404 定义

```

type multidata = record

```

```
(  
    contextualized_value_list:  
        array (0..*) of contextualized_data_type,  
),
```

描述

参见 11.4.3.2 中对记录和成分的数据描述。

11.5.3.3 contextualized_data_type

ISO/IEC 11404 定义

```
type contextualized_data_type = record
```

```
(  
    context_identifier:  
        characterstring, // SPM: 4096 characters  
    data:  
        reflit(unspecified),  
),
```

描述

参见 11.4.2.3 中对 record 和其成员属性的数据描述。

11.6 绑定的附加条款

无绑定的附加条款。

11.7 一致性附加条款

无一致性附加条款。

12 模块 12: multiple internationalized/localized strings and texts 的数据结构

12.1 模块简介

本章描述了多字符串和多文本：一组文本字符串可能具有不同的表示方式，但有相同的含义。多字符串可能被用于需要多种表现形式的应用程序中，如概念，而表示方式基于使用的语境会有所不同。这些差异可能包括国家、区域、文化、语言、交流方式（如视觉）、交流方法（如手语）、人种、环境等。多文本类似多字符串，多字符串将字符串基本数据类型用于所有其的表示中，而多文本支持使用不同的数据类型（如 JPEG 图像、WAV 声音文件）用于文本字符串的表示。

12.2 模块范围

本章描述了用于保存具有相同含义但有不同表示方式的文本字符串的数据结构。

12.3 功能

12.3.1 概述

第 10 章的 reflit 数据类型用于下述定义中。

12.3.2 多字符串数据结构

multistring 数据结构允许一个文本字符串根据不同的使用语境有不同的表示形式。

注：multistring 数据结构是类似于第 11 章中定义的 multivalued 数据结构，但存在以下差异：multistring 的定义是基于基本数据类型 characterstring，而 multivalued 使用用户定义的基本数据类型加以参数化。

示例：根据 ISO/IEC 总则第 2 章 1.1，“标准化”的概念在不同的语言中有相同的含义。这些不同的表示形式可以储存在多字符串中：

```
iso_iec_guide_2_standardization: multistring = // instantiates the multistring
(
contextualized_value_list =
( // strings in several languages all meaning "standardization"
////////////////////////////////////
( context_identifier = 'language="en"', // English text string
value_encoding = "ascii", value_as_text = "standardization", ),
////////////////////////////////////
( context_identifier = 'language="fr-CA"', // French Canadian text string
value_encoding = "iso-8859-1", value_as_text = "normalisation", ),
////////////////////////////////////
( context_identifier = 'language="ru"', // Russian text string
value_encoding = "iso-10646-1", value_as_text = "стандартизация", ),
////////////////////////////////////
( context_identifier = 'language="de"', // German text string
value_encoding = "iso-8859-1", value_as_text = "Normung", ),
////////////////////////////////////
( context_identifier = 'language="es"', // Spanish text string
value_encoding = "iso-8859-1", value_as_text = "normalización", ),
////////////////////////////////////
( context_identifier = 'language="it"', // Italian text string
value_encoding = "ascii", value_as_text = "normazione", ),
////////////////////////////////////
( context_identifier = 'language="nl"', // Dutch text string
value_encoding = "ascii", value_as_text = "nornalisatie", ),
////////////////////////////////////
( context_identifier = 'language="sv"', // Swedish text string
value_encoding = "ascii", value_as_text = "standardisering", ),
////////////////////////////////////
),
),
```

12.3.3 多文本数据结构

multitext 数据结构允许一个文本字符串根据不同的使用语境有不同的表示形式，同时，允许在单个的表示中有不同的表示形式。

注：multitext 数据结构与 multistring 数据结构类似，但有以下不同：multistring 使用文本字符串，表示为字符串，使用基本数据类型作为 multistring 数据结构实例中的每种表示形式，而 multitext 有用户定义的形式，数据类型以及在执行时确定的引用或值用于其每一种表示形式中。

示例：下面的例子说明了使用多数据数据结构来存储“number 2”概念的不同表示形式。

```
concept_of_number_2: multitext = // instantiates the multitext data structure
```

```
(
contextualized_value_list =
(
////////////////////////////////////
( // English text presentation (literal value stored)
context_identifier = 'language="en",mime-content-type=text/plain',
data = ( reflit_kind = literal, literal_value =
(
literal_kind = "text",
literal_encoding = "ascii",
literal_bintext = text,
literal_value_as_text = "two",
) ),
),
////////////////////////////////////
( // German HTML text presentation (reference to URL)
context_identifier = 'language="de",mime-content-type=text/html',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
reference_value_as_text = "http://zwei.org/zwei.html",
) ),
),
////////////////////////////////////
( // graphical presentation of numeral 2 (reference to GIF image)
context_identifier = 'mime-content-type=image/gif',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
reference_value_as_text = "http://zwei.org/2.gif",
) ),
),
////////////////////////////////////
( // German audio presentation of numeral 2 (reference to WAV file)
context_identifier = 'mime-content-type=audio/wav',
data = ( reflit_kind = reference, reference_value =
(
reference_kind = "url",
reference_bintext = text,
reference_value_as_text = "http://zwei.org/zwei.wav",
) ),
),
),
```

```

),
////////////////////////////////////
( // Morse Code presentation of numeral 2 (literal value stored)
context_identifier = 'mime-content-type=other/x-morse-code',
data = ( reflit_kind = literal, literal_value =
(
literal_kind = "text",
literal_encoding = "ascii",
literal_bintext = text,
literal_value_as_text = "..--",
) ),
),
////////////////////////////////////
),
),

```

12.4 抽象模型

12.4.1 概述

第 12 章对 multistring 数据结构和 multitext 数据结构的对象-模型加以描述。对象模型对应于 12.6 绑定独立的语法。

12.4.2 multistring

12.4.2.1 概述

图 4 是 multistring 和 contextualized_string 类的 UML 表示。

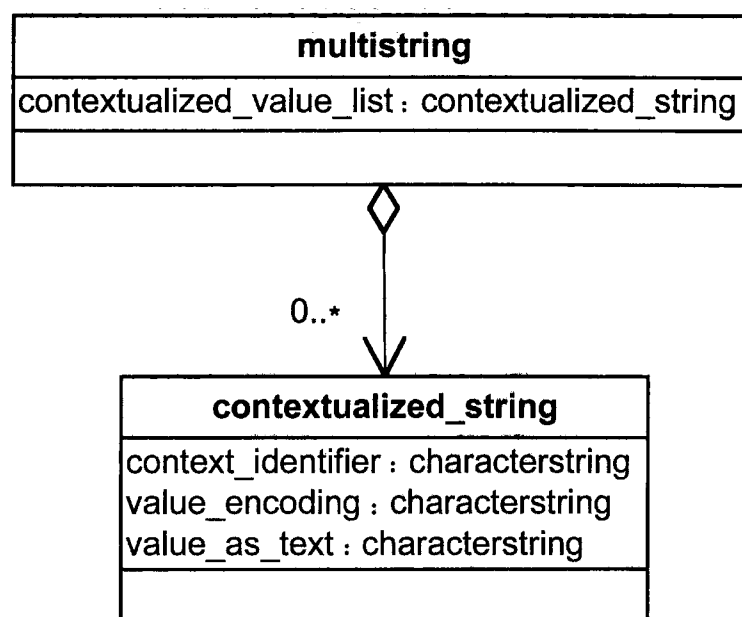


图 4 multistring, contextualized_string 的 UML 表示

12.4.2.2 multistring

multistring 类别包含如下的成员和属性：

——contextualized_value_list: contextualized_string 的数组(0..*): 值数组具有相同的意义, 每个

有自己的语境。

12.4.2.3 contextualized_string

contextualized_string 类别包含下列内容：

- context_identifier: characterstring: 标识符和语境相关, 如内容的描述。该数据元素是一个字符串, 有零个或多个下述空格分隔的成员：
 - "language='langcode'": 语言标识符, 由 IETF RFC 5646 加以定义。
 - "mode='interactionmode'": 交互模式, 定义为 ×××× (需要引用标准)。
 - "assertrequirements='specid'": 每个 specid 标准需求的声明 (需要 ISO URN 的 IETF)。
 - "certifyrequirements='specid;cert'": Certification of requirements, as per specification reference specid as certified by cert. (fix: editor's note, need IETF RFC for ISO URN)
 - "assertconformity='specid'": Assertion of requirements, as per specification reference specid. (fix: editor's note, need IETF RFC for ISO URN)
 - "certifyconformity='specid'": Certification of conformity, as per specification reference specid as certified by cert. (fix: editor's note, need IETF RFC for ISO URN)
 - " * ": Format specified external to this Clause.
- value_encoding: characterstring: 字符串编码方式。数据项允许有如下值：
 - "isoiec646": 使用 ISO/IEC 646 编码的字符串；
 - "isoiec8859-1": 使用 ISO/IEC 8859-1 编码的字符串；
 - "isoiec10646/utf-8": 使用 ISO/IEC 10646 UTF-8 编码的字符串；
 - "isoiec10646/utf-2": 使用 ISO/IEC 10646 UTF-2 编码的字符串；
 - "isoiec10646/utf-16": 使用 ISO/IEC 10646 UTF-16 编码的字符串, 包含未定义的八进制序；
 - "isoiec10646/utf-16BE": 使用 ISO/IEC 10646 UTF-16 编码的字符串, 包含低位数组后置的八进制序；
 - "isoiec10646/utf-16LE": 使用 ISO/IEC 10646 UTF-16 编码的字符串, 包含低位数组前置的八进制序；
 - "ascii": 使用 ANSI X3.4 编码的字符串；
 - value_encoding: characterstring: 用于独立表示的字符串。

示例: 以下是访问上述 UML 图中数据对象的例子:

```
self.contextualized_value_list
self.contextualized_value_list[0].context_identifier
self.contextualized_value_list[0].value_encoding
self.contextualized_value_list[0].value
self.contextualized_value_list[1].context_identifier
self.contextualized_value_list[1].value_encoding
self.contextualized_value_list[1].value
```

12.4.3 multitext

12.4.3.1 概述

图 5 是 multidata 和 contextualized_data 类的 UML 表示。

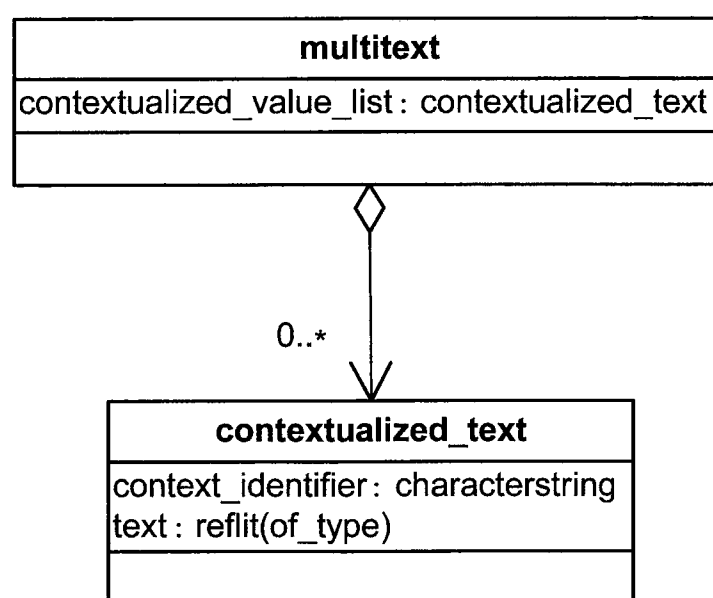


图 5 multitext, contextualized_text 的 UML 表示

12.4.3.2 multitext

multitext 类包含如下的成员和属性：

- contextualized_value_list: contextualized_string 的数组 (0..*): 值数组具有相同的意义, 每个有自己的语境。

12.4.3.3 contextualized_text

contextualized_text 类包含如下的成员和属性：

- context_identifier: characterstring: 标识符和语境相关, 如内容的描述。
- text: reflit(未指定): 与内容标识符相关的单个表示。reflit 数据结构的定义在第 10 章中。

示例: 以下是访问上述 UML 图中数据对象的例子:

```

self.contextualized_value_list
self.contextualized_value_list[0].context_identifier
self.contextualized_value_list[0].text // See footnote 19
self.contextualized_value_list[1].context_identifier
self.contextualized_value_list[1].text
  
```

12.5 计算描述和数据类型

12.5.1 概述

12.5 定义使用 ISO/IEC 11404 标记的数据类型。

12.5.2 multistring

12.5.2.1 概述

12.5.2 定义了 multistring 数据结构的数据类型。

12.5.2.2 multistring

在 ISO/IEC 11404 中的定义:

```

type multistring = record
(
  
```

contextualized_value_list:
array (0..*) of contextualized_string_type,
)

描述:

参见 12.5.2.2 record 及其成员的描述。

12.5.2.3 背景化_字符_类别

在 ISO/IEC 11404 中的定义:

```
type contextualized_string_type = record  
(  
  context_identifier:  
  characterstring, // SPM: 4096 characters  
  value_encoding:  
  characterstring, // SPM: 4096 characters  
  value_as_text:  
  characterstring,  
)
```

描述:

参见 12.5.2.3 记录及其成员的描述。

12.5.3 多文本

12.5.3.1 概述

12.5.3 定义了 multitext 数据类型的数据结构。

12.5.3.2 multitext

在 ISO/IEC 11404 中的定义:

```
type multitext = record  
(  
  contextualized_value_list:  
  array (0..*) of contextualized_text_type,  
)
```

描述:

参见 12.4.3.2 record 及其成员的描述。

12.5.3.3 contextualized_text_type

在 ISO/IEC 11404 中的定义:

```
type contextualized_text_type = record  
(  
  context_identifier:  
  characterstring, // SPM: 4096 characters  
  text:  
  reflit(unspecified),
```


),

描述:

见 12.4.2.3 record 及其成员的描述。

12.6 绑定的附加条款

无绑定的附加条款。

12.7 一致性附加条款

无一致性附加条款。

13 模块 13:slot tuple 的数据结构

13.1 模块介绍

第 13 章描述了元组项的数据结构,元组项通过使用 {identifier, kind, value} 3 元组的集合来描述数据值。第 14 章提供了一种数据结构来组合描述非结构化(或结构化)数据值表。

13.2 模块范围

13.3 功能

元组项数据结构提供了一个数据存储的单元,即,一个未预定义标识符和类型的数据项。元组项由下面的要素组成:标识符(如标签、标识符、名称等)、类型(如数据类型、类别等)以及其值(计算单元,比如, string, record, array, set, data-time, Boolean, integer 或 real)。

注:第 14 章提供了元组项的一个数据表,用来表示 11179-3 元模型以及专业扩展(11179-3 标准之外的数据)的部分内容。

示例:下面的例子是三个元组项(使用数据类型的 text-text-text 变种),每一个表示 E-mail 消息的一个要素:

```
e_mail_from_line: slot_tuple_as_ttt,
e_mail_from_line.identifier = "From",
e_mail_from_line.kind = "ietf/rfc822.from/string",
e_mail_from_line.value = "president@company.org",
e_mail_subject_line: slot_tuple_as_ttt,
e_mail_subject_line.identifier = "Subject",
e_mail_subject_line.kind = "ietf/rfc822.subject/string",
e_mail_subject_line.value = "Letter from the Company President",
e_mail_date_line: slot_tuple_as_ttt,
e_mail_date_line.identifier = "Date",
e_mail_date_line.kind = "ietf/rfc822.date/date",
e_mail_date_line.value = "2004-10-10T01:23:45.67",
```

IKV 元组可以被用在多种兼容的数据类型中,每一种都根据特定的使用情况加以优化。

13.4 抽象模型

13.4.1 概述

从概念上来说,元组项是一个 3 元组,表示一个标识符,类型和值。通常情况下,元组项用于组(参见第 14 章)中,同时用于普通的数据交换(存在数据类型在数据交换前可能未知的情况)。13.4 剩余的部分将描述元组项的对象模型。对象模型对应于 13.5 中绑定独立的语法。

13.4.2 元组项(slot_tuple)要素

slot_tuple 类是 {identifier, kind, value} 的三元组。slot_tuple 实例可以使用下面实现中的任意一个变种以适用于特定的用例。这个类包含如下的要素：

- 标识符(identifier)：多数据，与元组项实例的值关联的一个标签。选择标签的规则并不在国际标准的范围内。下面是标识符的几个例子：
 - "latitude"：元组项，对应于经纬度对中的纬度值(该标识符可以被存储在多数据中，或使用元组的字符串变种加以实现)；
 - "17" 或 17：标识符 17 (作为字符串或数字) 被用作索引(标签的一类)，其对应于元组数组中的第 17 个元组，如在时间序列数据集中的第 17 个数据点 (对应的值元素是时间序列数据本身)；
 - 0x13 0xFC 0x3D 0x8A：一个元组项，其标识符是一个内部指针或相关联的键，其指向对应的值(这个标识符可以被存储在多数据或以元组项八位字符串实现的形式加以存储)；
- 类别(kind)：多数据，与元组实例值相关的一个类别，选择类别的标准不在国际标准的范围之内；下面是类别的例子：
 - "iso_iec_11404.time(second,10,0)"：精度为 1 s 的时间戳；
 - "iso_iec_11404.scaled(10,2)"：包含 2 位小数的标定数值(如，货币的数据类型为美元或美分)；
 - "java.String"：Java string 类；
 - "uml.Integer"：UML 中的整型；
 - "binary"：二进制数据；
- value：多数据，值本身，具有上述的标识符和类别，如下是值的一些例子：
 - "20020317T223000-0500"：日期数据，表示 三月 17，10:30 下午，东部时间；
 - 314.15：表示 €314.15；
 - "стандартизация"：Java String 类实例，"standard"的俄语表示；
 - 2147483647：UML 整型，表示 $2^{31} - 1$ ；
 - 0x00 0x01 0x02 0x03 0x04：5 个八位二进制值。

13.4.3 元组项及其变种

元组项可能被用于多种兼容的数据类型，每一种都根据特定的使用情况加以优化，见表 1。

表 1 元组项中的数据类型

变种	元组项中的数据类型			说明
	标识符	类型	值	
slot_tuple	Multidata	multidata	multidata	最通用的数据类型。所有其他的 IKV 变种基于本数据类型
slot_tuple_as_ttt	characterstring	characterstring	characterstring	用于 text-only 的应用中
slot_tuple_as_ttrl	characterstring	characterstring	reflit	Text only + reflit "value"
slot_tuple_as_ttmd	characterstring	characterstring	multidata	Text-only + multidata "value"
slot_tuple_as_bbb	Octetstring	octetstring	octetstring	用于 binary-only 的应用中
slot_tuple_as_btb	Octetstring	characterstring	octetstring	具有 text 类型的 Binary 应用
slot_tuple_as_btmd	Octetstring	characterstring	multidata	具有 text 类型以及 multidata 值的 Binary 应用

示例：下面是访问 UML 图中数据对象的例子：

self.identifier

self.kind

self.value

13.4.4 slot_tuple

图 6 是 slot_tuple 的 UML 表示。

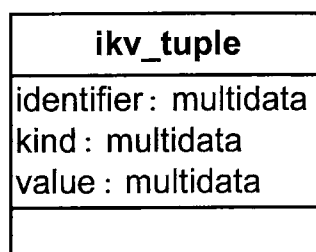


图 6 slot_tuple 数据类型的 UML 表示

13.4.5 slot_tuple_as_ttt

图 7 是 slot_tuple_as_ttt 类的 UML 表示。

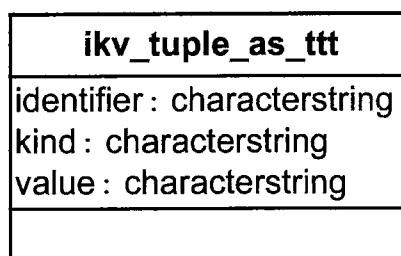


图 7 slot_tuple_as_ttt 数据类型的 UML 表示

13.4.6 slot_tuple_as_ttrl

图 8 是 slot_tuple_as_ttrl 类的 UML 表示。

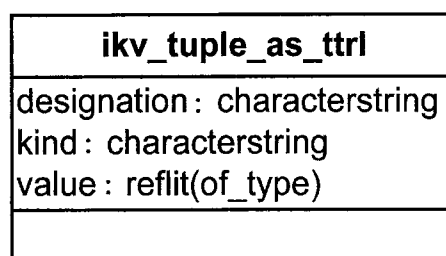


图 8 slot_tuple_as_ttrl 数据类型的 UML 表示

13.4.7 ikv_tuple_as_ttmd

图 9 是 slot_tuple_as_ttmd 类的 UML 表示。

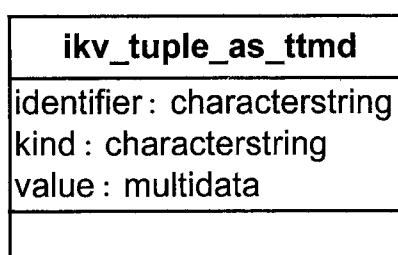


图 9 slot_tuple_as_ttmd 数据类型的 UML 表示

13.4.8 ikv_tuple_as_bbb

图 10 是 slot_tuple_as_ttt 类的 UML 表示。

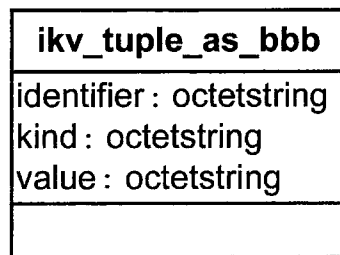


图 10 slot_tuple_as_bbb 数据类型的 UML 表示

13.4.9 ikv_tuple_as_btb

图 11 是 slot_tuple_as_ttt 类的 UML 表示。

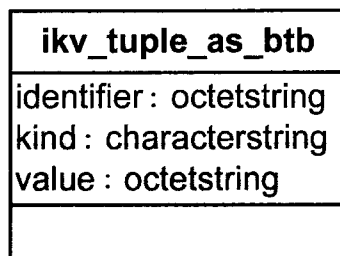


图 11 slot_tuple_as_btb 数据类型的 UML 表示

13.4.10 ikv_tuple_as_btmd

图 12 是 slot_tuple_as_ttt 类的 UML 表示。

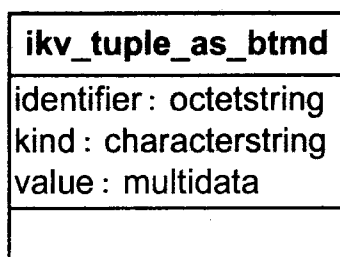


图 12 slot_tuple_as_btmd 数据类型的 UML 表示

13.5 计算描述和数据类型

13.5.1 概述

13.5 使用 ISO/IEC 11404 符号来定义数据类型。嵌入到 ISO/IEC 11404 注释中的条款是规范化的。

13.5.2 数据类型

13.4 定义的数据类型用于 IKV 元组的变种。如下是 ISO/IEC 11404 符号中数据类型的定义。

在 ISO/IEC 11404 中的定义：

```

type slot_tuple = record
(
identifier: multidata, kind: multidata, value: multidata,
),
    
```

```

type slot_tuple_as_ttt = record
(
identifier: characterstring, kind: characterstring, value: characterstring,
),
type slot_tuple_as_ttrl = record
(
identifier: characterstring, kind: characterstring, value: reflit(kind),
),
type slot_tuple_as_ttmd = record
(
identifier: characterstring, kind: characterstring, value: multidata,
),
type slot_tuple_as_bbb = record
(
identifier: octetstring, kind: octetstring, value: octetstring,
),
type slot_tuple_as_btb = record
(
identifier: octetstring, kind: characterstring, value: octetstring,
),
type slot_tuple_as_btmd = record
(
identifier: octetstring, kind: characterstring, value: multidata,
),

```

13.6 绑定的附加条款

无绑定的附加条款。

13.7 一致性附加条款

无一致性附加条款。

14 模块 14: unstructured table of slot tuples 的数据结构

14.1 模块介绍

第 13 章描述了 identifier-kind-value 元组项数据结构,其通过使用 3 元组 {identifier, kind, value} 来描述单个数据值。

第 14 章提供了组合特定数据值表的数据结构,标识符作为一个关键字用来访问表中的一行。

14.2 模块范围

第 14 章提供了组合特定数据值表的数据结构。其基于第 13 章中定义的 identifier-kind-value 元组项数据结构。

14.3 功能

元组项表数据结构允许表示非结构化的元组项表。本章不涉及数据元组的结构(如,名字的选择以及他们之间的嵌套关系)。外部的标准可以基于本标准提供数据元组需要的数据结构。

示例:下面的例子来自 11179-3 元模型。

```
example_11179_data_element: slot_tuple_table_as_ttt =
(
(
"data_element_administration.creation_date",
"time",
"2004-10-31"
),
(
"data_element_administration.administrative_note",
"string",
    "An example note."
),
(
"representation_class_qualifier",
"characterstring",
"quantity"
),
(
"data_element_precision",
"integer",
"2"
),
(
"exemplified_by.0.data_element_example_item",
"characterstring",
"9.00"
),
),
),
```

14.4 抽象模型

14.4.1 概述

从概念上讲,一个 3 元组元组项表示一个标识符,一个类型,以及一个值(定义在第 13 章中)。通常情况下,元组项用在组中(定义在本章中)同时,用于一般的数据交换中(数据类型在数据交换之前可能未知)。本章剩余的部分描述了 IKV 元组表及其变种的对象模型。

14.4.2 slot_tuple_table 及相关类

图 13 是 slot_tuple_table 类的 UML 表示形式。

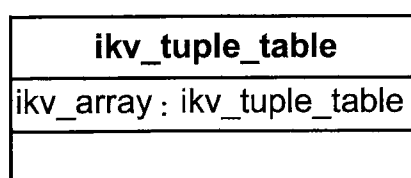


图 13 slot_tuple_table 数据类型的 UML 表示

除了 slot_tuple_table 之外,还有一些类似的 UML 类,如 slot_tuple_table_as_ttt, slot_tuple_table_as_ttrl, slot_tuple_table_as_ttmd, slot_tuple_as_bbb, slot_tuple_as_btb, slot_tuple_as_btmd。

14.5 计算描述和数据类型

14.5.1 概述

14.5 使用 ISO/IEC 11404 符号定义了数据类型。

14.5.2 数据类型

14.4 定义了每个元组项变种的数据类型。下面是使用 ISO/IEC 11404 符号数据类型的总结。

在 ISO/IEC 11404 中的定义:

```

type slot_tuple_table = array (0..*) of slot_tuple,
type slot_tuple_table_as_ttt = array (0..*) of slot_tuple_as_ttt,
type slot_tuple_as_ttrl = array (0..*) of slot_tuple_as_ttrl,
type slot_tuple_table_as_ttmd = array (0..*) of slot_tuple_as_ttmd,
type slot_tuple_table_as_bbb = array (0..*) of slot_tuple_as_bbb,
type slot_tuple_table_as_btb = array (0..*) of slot_tuple_as_btb,
type slot_tuple_table_as_btmd = array (0..*) of slot_tuple_as_btmd,

```

14.6 绑定的附加条款

无绑定的附加条款。

14.7 一致性附加条款

无一致性附加条款。

15 模块 15:reified relationships and relationshipssystems 的数据结构

15.1 模块简介

此模块指定的数据结构用于具体化对象之间的关系。

15.2 模块范围

第 15 章提供了用于表示具体化关系和关系系统的数据的数据结构。

15.3 功能

reified_relationship 和 reified_relationship_system 数据结构支持具体化关系,即将关系转化为数据,同时支持具体化关系系统。例如,数据结构可以用来表示任意的对象集合之间的关系。

15.4 抽象模型

15.4.1 概述

从概念上来讲,具体化关系是一种关系类型(例如,一个名称或标识符),以及所有的对象列表,每个对象都有它自己的角色。15.4 的剩余部分描述了对象模型。

15.4.2 reified_relationship_system 和 reified_relationship

图 14 是 reified_relationship_system, reified_relationship 以及 object_role_pair 类的 UML 表示形式。

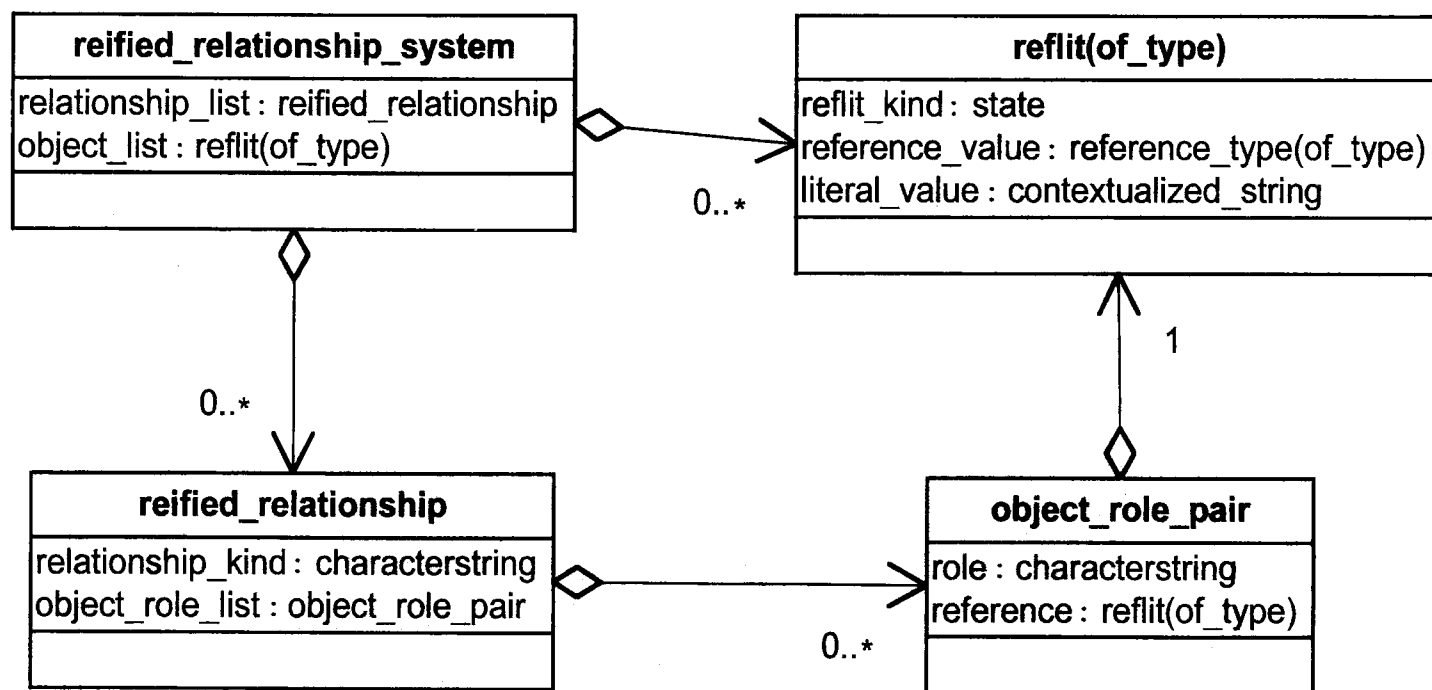


图 14 具体化关系系统的 UML 陈述

reified_relationship_system 类包含如下的成员和属性:

- relationship_list: array (0..*) of reified_relationship: 关系的列表;
- object_list: array (0..*) of replit: 对象的列表。

reified_relationship 类包含如下的成员和属性:

- relationship_kind: characterstring: 一个用来表示类型,名称或者关系类型标识符的字符串;
- object_role_list: array (0..*) of object_role_pair: 对象及其角色的列表。

object_role_pair 类包含如下的成员和属性:

- role: characterstring: 一个用来表示类型,名称或者对象角色标识符的字符串;
- reference: replit: 指向对象的指针。

15.5 计算描述和数据类型

15.5.1 概述

15.5 定义了使用 ISO/IEC 11404 符号定义了数据类型。

15.5.2 reified_relationship_system

在 ISO/IEC 11404 中的定义:


```

type reified_relationship_system = record
(
relationship_list:
array (0..*) of reified_relationship,
object_list:
array (0..*) of reflit,
),
描述:
参见 15.4.2 中记录及其成员属性的描述。

```

15.5.3 reified_relationship

在 ISO/IEC 11404 中的定义:

```

type reified_relationship = record
(
relationship_kind:
characterstring,
object_role_list:
array (0..*) of object_role_pair,
),
描述:
参见 15.4.2 中描述了记录及其成员属性。

```

15.5.4 object_role_pair

在 ISO/IEC 11404 中的定义:

```

type object_role_pair = record
(
role:
characterstring,
reference:
reflit,
),
描述:
参见 15.4.2 中记录及其成员属性的描述。

```

15.6 绑定的附加条款

无绑定的附加条款。

15.7 一致性附加条款

无一致性附加条款。

16 模块 16: UPU postal data 的数据结构

16.1 模块的介绍

第 16 章描述的数据结构对应于万国邮联 (UPU) S42a-6 国际邮政地址要素及其模板—A 部分: 概

念层次关系及其模板语言。UPU S42 系列指定了对应邮政地址要素以及邮政地址片段的数据要素的结构和语法。

注：尽管邮政地址在 11179-3 原模型中使用，其也可以在更为宽泛的内容数据的语境中加以使用，参见第 19 章。

16.2 模块范围

第 16 章提供了邮政通讯地址的数据结构描述。UPU S42 地址要素作为构造邮政地址的基础。

16.3 功能

postal_address 数据结构包括 1)普通的邮政地址的国际化结构，以及 2)以一种或多种国家具体格式的表现形式。第 16 章可以被用于表示原始的邮政地址，或者以单个国家指定格式的邮政地址，或者两者。

16.4 抽象模型

16.4.1 概述

16.4 的剩余部分是 postal address 对象模型的描述。

16.4.2 邮政地址

16.4.2.1 postal_address_class

postal_address_class 由原始的邮政地址和转换的邮政地址所组成，在零个或更多的上下文的使用中可以被文本化。详见图 15。

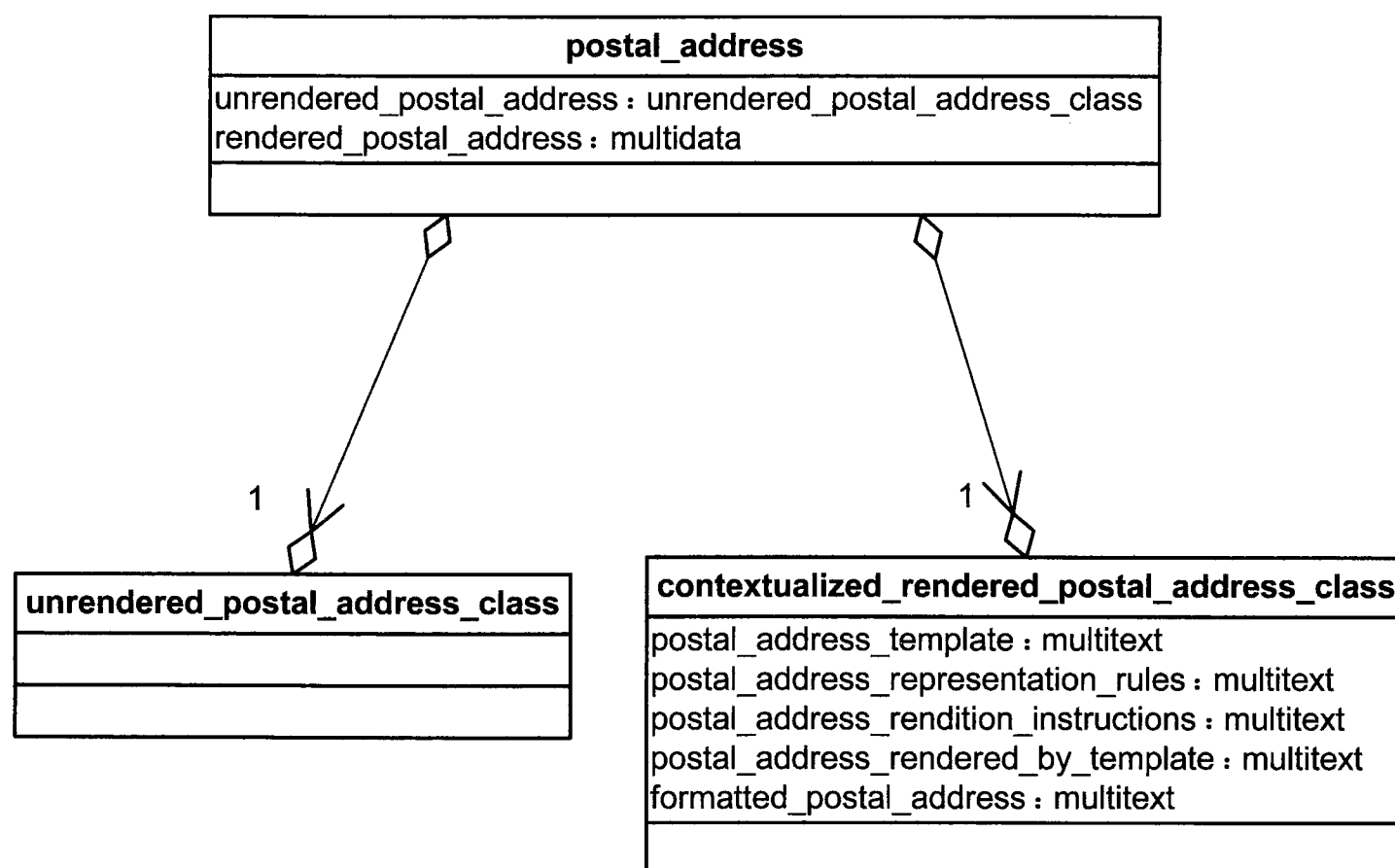


图 15 邮政地址结构的 UML 表示

16.4.3 原始的邮政数据

16.4.3.1 原始的邮政地址结构

邮政地址由段组成；段由结构以及/或邮政地址要素组成；结构由邮政地址要素组成。见图 16。

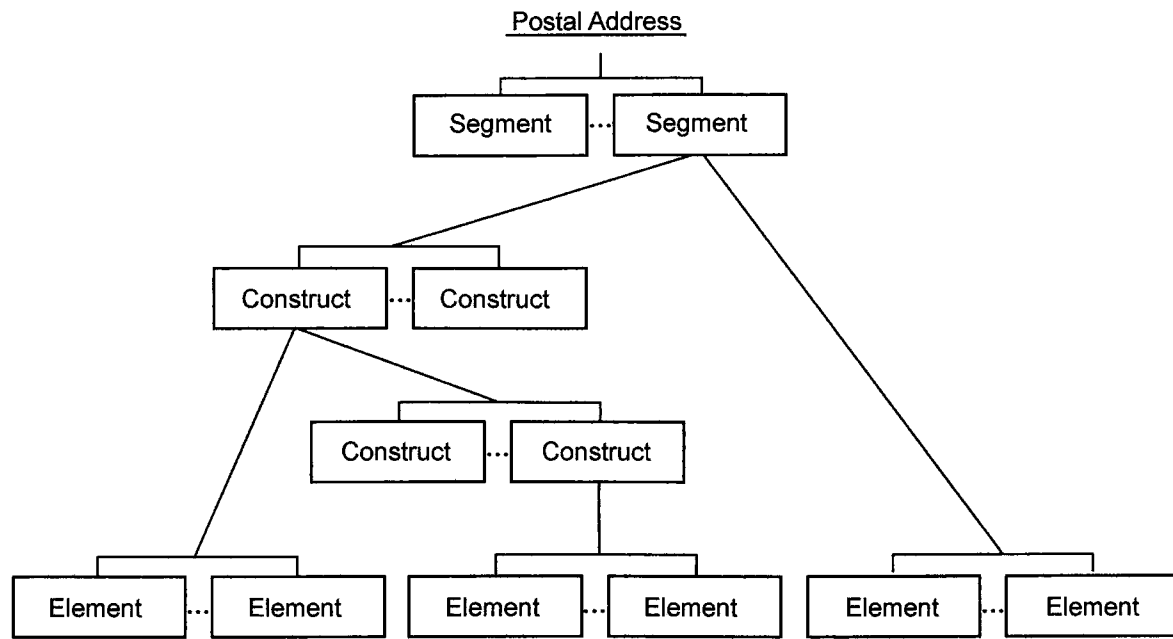


图 16 邮政地址结构 [本图来源于 UPU S42]

段可以认为是更低的层次结构。见图 17。

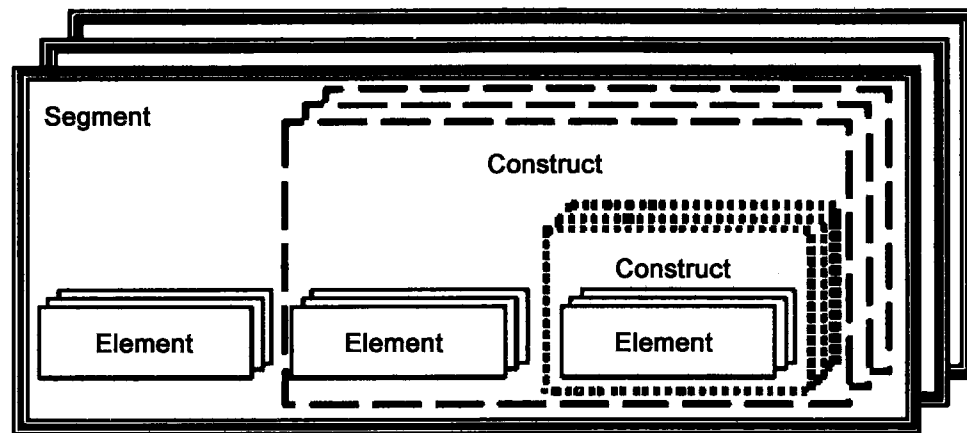


图 17 段,结构以及邮政地址要素的关系

下面的 UML 图表示上述邮政地址结构。见图 18。

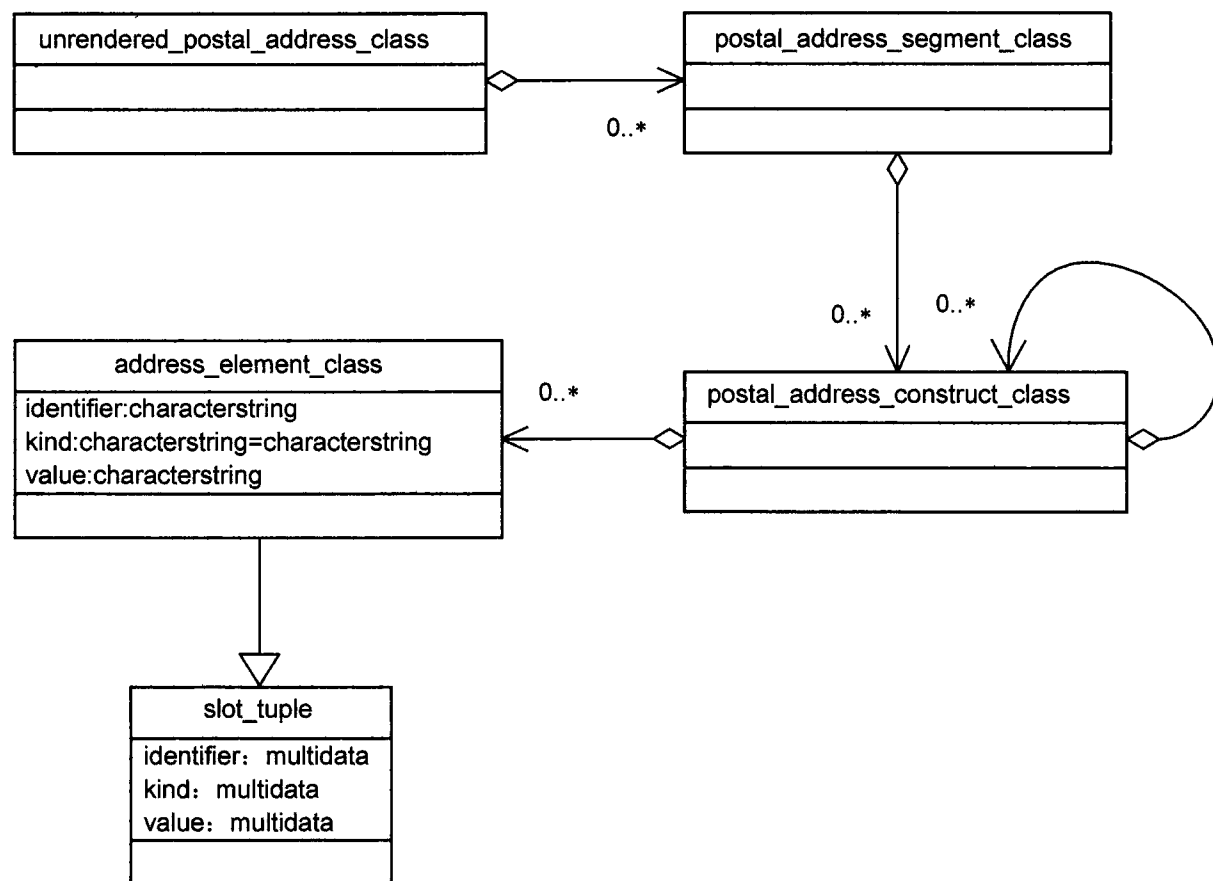


图 18 类的 UML 表示形式:原始的邮政地址,邮政地址段,邮政地址结构,地址要素

16.4.3.2 邮政地址要素

原始的邮政地址由 4 个邮政地址段组成：寄件人标准(可选)、收件人标准(可选)、邮件收件人分发信息(可选)、收件地点标准(必须)。见图 19。

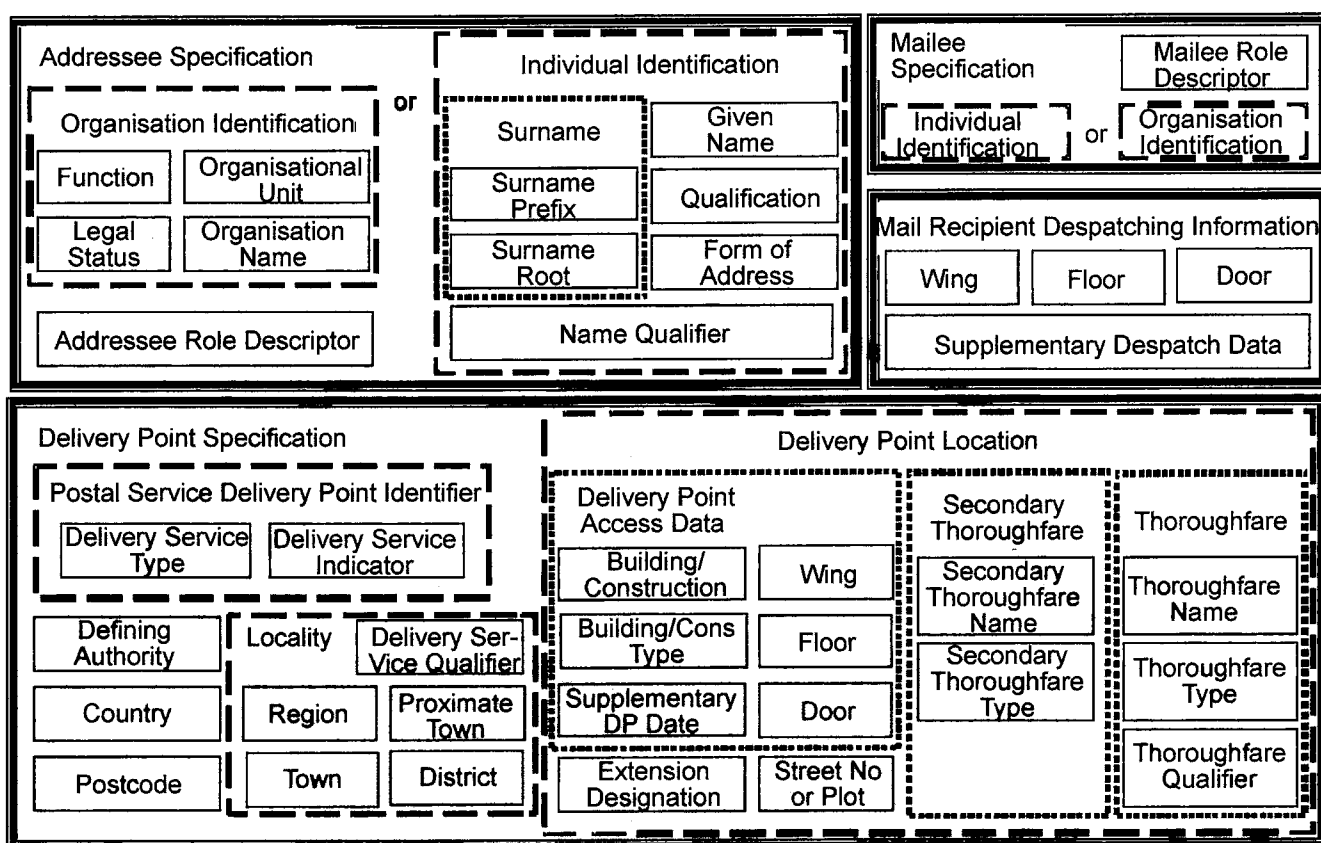


图 19 邮政地址——所有要素 [图来源于 UPU S42]

16.4.3.3 邮政地址元素

原始的邮政地址是由要素组成的嵌套的结构。下面的代码对应一级、二级以及三级的数据标识符。

根据其定义,邮政地址要素是生成地址所需的基本的概念单元。一个要素可以在地址的不同位置多次表示。比如,一个国家中的地址可能使用不同的位置,在地址要素的翻译上可能不同,或者不同时出现。另外,地址要素,比如区域,可能在相同的地址有多次出现。在 UPU S42 中要素的多次串叫做实例,不同于要素的一部分。单个要素的不同部分常常需要在翻译程序中加以区分,比如,为了插入合适的标点。实例和部分由两个主要的子元素组成,每一个都表示一个或两个要素子类型码的数位。要素子类型用来帮助表示这些多个实例和部分,同时方便的创建地址模板。当要素只表示一次,同时与地址不可分割时,我们不需要多个实例或多个部分,同时,不会使用子类型,其要素需要直接的包含在模板中。因此要素以及要素的子类型可以一起认为是模板中的组成块。

在组成要素子类型中,有一个与次数相关的问题。要素比如 district/sector 可能包含很多级别,位置多次出现在地址中。一旦明确子类型的需求,要么可以限制其出现的次数,或者使用无限的次数。定义无限次数的区域级别并不困难,比如使用 XML 模式就可以实现,但是在实际的使用中,针对单个邮政地址,其可以加以限制。而且定义无限的次数对设计地址数据库也没有太多好处。由于这些原因,元素子类型在 S42 中这样定义,提供足够的实例和部分来处理已知的问题,在实现中也提供了一些扩展性。

16.4.3.4 要素编码

要素编码由两个层次级别组成,前两个数字表示段/结构,后两个数字表示在段/结构中的要素,两者之间使用点号加以分割:xx.yy。

下面的标识符用在段层次中：

- 10 收件人标准段
- 20 寄件人标准段
- 30 邮件收件人分发信息段
- 40 收件地点标准段

段编码的第二个数字保留用于复制段，即提供段中所有要素以及子类型的副本。地址的形式类似于 10.05，表示编码为 11.05 的一个段副本。这种能力使得多个收件人可以再一个邮政地址中表示，同时存在多个邮政地址的一个收件人也可以加以表示。这样一来，就方便了地址数据库的设计。但是，很多邮局规定只存在一个收件点。一些邮政机构运行在相同的邮政区，一个邮箱与道路地址相关联；比如南非。

段级别的第二位数字限制了最多有 10 个段，其中的 4 个已经定义了。在要素级别，代码使用两个数字表示每个要素，允许最大 100 个要素。对于 S42，34 个要素以及 98 个子类型已经被定义了。这种方法保证要素列表相对紧凑，提供了在存储、组合以及转换邮政地址要素中更高级的粒度和不同点。

下面的标识符被分配到要素级：

- 00 组织名
- 01 法律地位
- 02 组织单元
- 03 功能
- 04 地址角色描述符
- 05 地址形式
- 06 名
- 07 姓氏前缀
- 08 姓氏
- 09 名称标识符
- 10 条件
- 11 寄件人角色描述符
- 12 定义的权限
- 13 邮政编码
- 14 国家名
- 15 区域
- 16 村镇
- 17 区
- 19 运输服务标识符
- 20 其他运输服务标识符
- 21 道路
- 24 街道号
- 26 建筑
- 28 扩展名
- 29 栋
- 30 单元
- 31 楼层
- 32 门号
- 33 其他的补充信息

34 补充的收件地点数据

35 运输服务标识符

41 国家编码

43 多国区域

44 国际化路径信息

元素的子类型编码由子类型元素的编码加连接符和一个单数字标识符,再加一个连接符和一个单数字标识符组成。如果子类型隐含在维度信息中,其使用 0 表示,否则的话使用 1~9 中的一个值加以表示。作为保留规则,当在模板中直接使用元素时,格式 xx.yy 和格式 xx.yy-z-z(z=0)表示相同的含义。

注:元素的子类型“door type”使用 40.32-0-1 表示,当其描述投递点规范的相关信息时。元素定义了门的一部分。但是相同的元素子类型,在收件人调度信息中被定义为 30.32-0-1。

示例:40.13-0-1 是 40.13 的一部分。40.17-1-0 是 40.17 的实例。40.17-1-1 以及 40.17-1-2 是该实例的部分。40.17-2-0 是 40.17 的第二个实例。40.13-0-0 等于 40.13。

16.4.3.5 元组项的元素数据标识符

对于使用在元组项中的邮政地址要素,标识符应包含下列格式:

<prefix> _ <postal_element_identifier> _ <element_name>

<prefix>是字符串“int_upu_s42”,<postal_element_identifier>在 16.4.3.4 中加以描述,使用下划线来替换所有的符号,同时<element_name>在 16.4.3.4 中加以描述。下述是 16.4.3.4 中的邮政要素的元组项标识符:

int_upu_s42_10_06_1_addressee_given_name_part_1

int_upu_s42_12_31_1_recipient_dispatching_door_type

16.4.4 文本化的邮政地址

此类保存文本化的邮政地址,零个或者多个文本化的邮政地址可能被包含在一个原始的的邮政地址中。

16.4.4.1 Contextualized_rendered_postal_address

contextualized_rendered_postal_address 类保存单个文本化的邮政数据。该类包含如下的成员属性:

- postal_address_template: multitext,这个要素保存邮政地址模板,定义在 UPU S42a-6 中;
- postal_address_representation_rules: multitext,这个要素定义了数据完整性和数据转换规则,定义在 UPU S42a-6 中;
- postal_address_rendition_instructions: multitext,这个要素定义了翻译指令,定义在 UPU S42a-6 中;
- postal_address_redenred_by_template: multitext,该要素定义了地址的格式,作为由翻译指令处理的文本话的邮政地址。

示例 1: France 模板实例

\Line 1\
 [form of address] [qualification] [given name] \surname\
 OR
 \organization name\ [legal status]
 [Line 2] [addressee role descriptor] [mailee role descriptor] [form of address] [qualification] [given name]

[surname]
 OR
 [addressee role descriptor] [mailee role descriptor] \organisation name\ [legal status] [organisational unit]
 [function]
 OR
 [floor] [door] [supplementary dispatch data]
 \Line 3\ [wing] [floor] [door] [supplementary dispatch data]
 OR
 [building/construction type] [building/construction] [wing] [floor] [door] [supplementary delivery point data]
 [Line 4] \street no. or plot\ [extension designation] [thoroughfare type] \thoroughfare name\
 \Line 5\ \delivery service type\ \delivery service indicator\ [town]
 [Line 6] \postcode\ \town\ [region] [district] [delivery service qualifier]
 \Line 7\ \country\.

示例 2: France 的表示规则实例

1. In a geopostal address include the delivery point location information if it is known.

示例 3: France 的翻译指令

1. An address has a maximum of 6 lines (7 for international mail).
2. Only lines that contain information will be printed. If the optional lines are not used there will be no extra space between the lines.
3. An address line has a maximum of 38 characters (including spaces).
4. There must be a space between two words.
5. No punctuation marks, underlining or words in italic are allowed in delivery point location line.
6. Line 6 must always be printed in capital letters and preferably lines 4 to 6.
7. The address blk must be justified to the left.
8. In street numbers 5 bis or 5 ter is shortened to 5 B or 5 T.
9. In the event of a street number with 2 separate sets of figures, only the first number is indicated in line 4 (4 to 8 should written as 4 and 12/14 should be written as 12).

示例 4: 格式化的地址示例

Formatted address Address elements
 MONSIEUR JEAN DURAND 10.05 MONSIEUR
 25 RUE DES FLEURS 10.06 JEAN
 33500 LIBOURNE 10.08 DURAND
 14.24 25
 14.22 RUE
 14.21 DES FLEURS
 13.13 33500
 13.16 LIBOURNE

16.5 计算的数据描述和数据类型

16.5.1 概述

16.5 描述了使用 ISO/IEC 11404 符号定义的数据类型。

16.5.2 邮政-地址

在 ISO/IEC 11404 中的定义:

```
type postal_address = record
(
  unrendered_postal_address:
    unrendered_postal_address_class,
  rendered_postal_address:
    multidata, // multidata contains contextualized_unrendered_postal_class
),
```

描述:

参见 16.4.2 中对记录及其成员属性的描述。

16.5.3 unrendered_postal_address_class

在 ISO/IEC 11404 中的定义:

```
type unrendered_postal_address_class = record
(
  postal_element:
    array (0..*) of ikv_tuple,
),
```

描述:

参见 16.4.3 中对记录及其成员属性的描述。

16.5.4 contextualized_rendered_postal_address_class

在 ISO/IEC 11404 中的定义:

```
type contextualized_rendered_postal_address_class = record
(
  postal_address_template:
    multitext,
  postal_address_representation_rules:
    multitext,
  postal_address_rendition_instructions:
    multitext,
  postal_address_rendered_by_template:
    multitext,
  formatted_postal_address:
    multitext,
),
```

描述:

参见 16.4.4 中对记录及其成员属性的描述。

16.6 绑定的附加条款

无绑定的附加条款。

16.7 一致性附加条款

The following is the 16-21090 profile mapping. In 21090 subclause 7.7.1.1, the data elements map to

the 19773 module 16 postal data:

- ↓ |21090 AL (address line) corresponds to item 34 (supplementary delivery point data)
- ↓ |21090 ADL (additional locator) corresponds to secondary items 26 (building/construction), 28 (extension identifier), 29 (wing).
- ↓ |21090 UNID (unit identifier) corresponds to secondary items 31 (floor), 32 (door)
- ↓ |21090 DAL (delivery address line) corresponds to secondary item 19 (delivery service indicator).
- ↓ |21090 DINST (delivery installation type) corresponds to secondary item 20 (alternate delivery service indicator).
- ↓ |21090 DINSTA (delivery installation area) corresponds to secondary item 16 (town)
- ↓ |21090 DINSTQ (delivery installation qualifier) corresponds to secondary item 35 (delivery service qualifier)
- ↓ |21090 DMOD (delivery mode) corresponds to secondary item 19 (delivery service indicator).
- ↓ |21090 DMODID (delivery mode identifier) corresponds to secondary item 19 (delivery service indicator)
- ↓ |21090 SAL (street address) corresponds to secondary items 21 (thoroughfare name), 22 (thoroughfare type), 23 (thoroughfare qualifier)
- ↓ |21090 BNR (building number) corresponds to secondary item 24 (street number or plot).
- ↓ |21090 BNN (building number numeric) corresponds to the numeric portion of secondary item 24 (street number or plot)
- ↓ |21090 BNS (building number suffix) corresponds to the non-numeric portion of secondary item 24 (street number or plot)

17 模块 17:ITU-TE.164 电话号码数据的数据结构

17.1 模块介绍

第 17 章对应于 ITU-TE.164 电话号码数据的数据结构。数据结构可以储存国际编号方案(如 +12125551212) 或私人的编号方案(如 0800990011)的电话号码;电话号码可以包括内部分线交换机(私人电话交换机)分机扩展(如 +12125551212+234、0800990011+234)。

注：“+”是一个电话号码的结构化要素,如在“0800990011+234”中“0800990011”和“234”是完整电话号码中的独立的子部分。

17.2 模块范围

本章提供了电话号码的数据结构的描述,其可能兼容 ITU-T E.164 标准。数据结构可以储存国际编号方案(如 +12125551212)或私人的编号方案(如 0800990011)的电话号码;电话号码可以包括内部分线交换机(私人电话交换机)分机扩展(如 +12125551212+234、0800990011+234)。

17.3 模块功能

phone_number 数据结构包括电话号码的要素。电话号码可以根据使用的不同情况分为分解为相同表示项。

示例 1:电话号码可以用+12125551212 等效表示(要素类型在括弧[]内):

E.164 格式:+12125551212 [E.164 字符串]

E.164 带标点:+1 212 555 1212 [带标点的 E.164 字符串]

E.164 分解:+12125551212 [E.164 字符串]

美国国家标准格式:(212) 555-1212 [无结构字符串]

纽约地方性格式:411 [无结构字符串]

法国访问格式:00 [地方性国际通用前缀(地区号码)]

1 [国家代码]

212 [城市代码]

555 1212 [本地号码]

示例 2:上述结构将被保存在如下电话要素结构中:

```
nyc_directory_assistance: phone_number_class =
(
  phone_number_element_string = (
    ( contextualized_value_list =
      ( context_identifier = "ITU-T E.164",
        value = (
          ( "ITU-T E.164", "iso-646", "+12125551212" ),
        )
      )
    ),
    ( contextualized_value_list =
      ( context_identifier = "ITU-T E.164 with punctuation",
        value = (
          ( "formatted_string", "iso-646", "+1 212 555 1212" ),
        )
      )
    ),
    ( contextualized_value_list =
      ( context_identifier = "ITU-T E.164 decomposed",
        value = (
          ( "international_numbering_plan_prefix", "iso-646", "+" ),
          ( "country_code", "iso-646", "1" ),
          ( "city_code", "iso-646", "212" ),
          ( "local_number", "iso-646", "5551212" ),
        )
      )
    ),
    ( contextualized_value_list =
      ( context_identifier = "US",
        value = (
          ( "formatted_string", "iso-646", "(212) 555-1212" ),
        )
      )
    ),
    ( contextualized_value_list =
      ( context_identifier = "US-NY-newyork",
        value = (
          ( "formatted_string", "iso-646", "411" ),
        )
      )
    ),
    ( contextualized_value_list =
      ( context_identifier = "FR",
```

```

value = (
    ( "international_prefix", "iso-646", "00" ),
    ( "country_code", "iso-646", "1" ),
    ( "city_code", "iso-646", "212" ),
    ( "local_number", "iso-646", "5551212" ),
)
)
),
)

```

17.4 抽象模型

17.4.1 概述

17.4 的剩余部分描述了 phone_number 的对象模型。图 20 是 UML 对象模型。

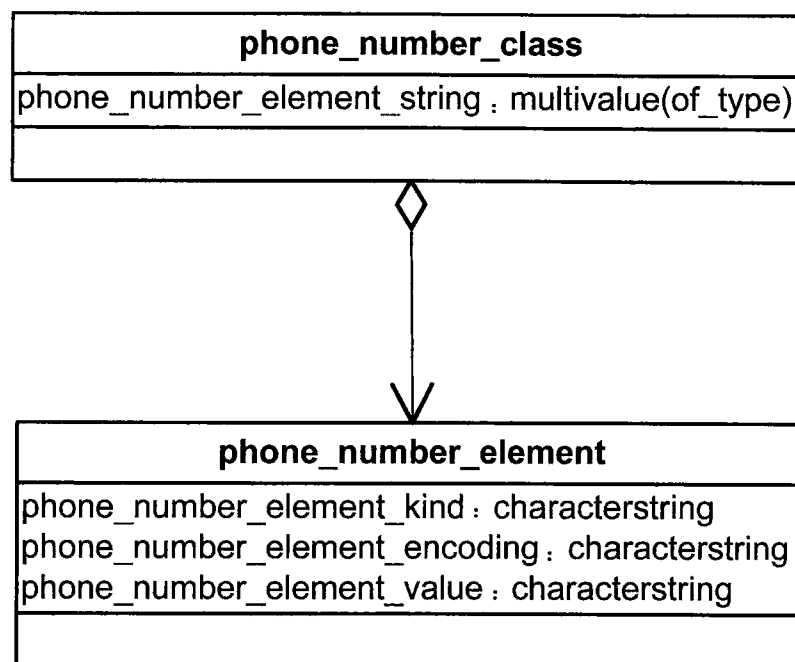


图 20 电话号码结构的 UML 表示

17.4.2 phone_number_class

phone_number_class 由一系列电话号码要素组成,其可能包含一个或多个使用场景。这个类包含如下的要素:

- phone_number_element_string: multidata(array of phone_number_element): 此要素是电话号码数组。

17.4.3 phone_number_element

postal_number_element 保存了电话号码要素的单个表示形式的一部分。这个类包含如下的属性与成员。

- phone_number_element_kind: characterstring, 电话号码要素的类型。
- phone_number_element_encoding: characterstring, 电话号码要素值的编码类型。
- phone_number_element_value: characterstring, 部分电话号码要素的文本表示。

注: 很多种表现形式可能都为同样的电话号码,比如通过国际编号方案表示(" +12125551212 ")和通过一些私人的编号方案(" 2125551212 "," 411 "," 0012125551212 ")来表示。表示形式可能是有格式的(如"(212)555 - 1212)"," +1 212 555 1212")或表示形式可能会被分解成元素(如把电话号码间断成部分:" +","1","212",

" 5551212 ")。始于元素类型 international_numbering_plan "(有格式的)或"international_numbering_plan_prefix "(离解)的电话号码表示方法要在 ITU - T E.164 格式中;否则、电话号码就是私人的编号方案。如果有格式的字符串包括" + "特性也就是说非第一特性或如果离解电话号码包括"扩展名"电话号码元素,则电话号码包括扩展名。

示例 1:电话号码要素类型示例:

国际编号方案前缀
 国家代码
 城市代码
 本地号码
 扩展名

示例 2:电话号码要素编码示例:

iso-646
 ascii

示例 3:电话号码要素值示例:

+
 1
 212
 5551212

17.5 计算的数据描述和数据类型

17.5.1 概述

17.5 使用 ISO/IEC 11404 定义了使用的数据类型。

17.5.2 phone_number_class

在 ISO/IEC 11404 中的定义:

```
type phone_number_class = record
(
    phone_number_element_string:
        multivalue(array (0..*) of phone_number_element),
),
```

描述:

参见 17.4.2 中对记录和成分的数据描述。

17.5.3 电话-号码-元素

在 ISO/IEC 11404 中的定义:

```
type phone_number_element = record
(
    phone_number_element_kind:
        characterstring,
    phone_number_element_encoding:
        characterstring,
    phone_number_element_value:
        characterstring,
),
```

描述:

参见 17.4.3 中对记录和其成员属性的描述。

17.6 绑定的附加条款

无绑定的附加条款。

17.7 一致性附加条款

无一致的附加条款。

18 模块 18: who-what-where-when-why-how (W5H) event data 的数据结构

18.1 模块介绍

第 18 章描述了用于存储谁-什么-哪里-何时-那相当于那什么-怎样(W5H)事件数据的数据结构。

注 1: 空间性以及时态性的特点还有几何与拓扑的特性与这两个标准中的描述是一致的: ISO 19107 地理信息——空间模式以及 ISO 19108 地理信息——时态模式。在这个国际标准中定义的模型里,空间特征由一个或多个空间属性加以描述,空间属性使用几何对象(GM_Object)或拓扑对象(TP_Object)来表示。几何对象使用量化描述方法,使用坐标和数学函数来描述诸多空间特征,包括维度、位置、大小、形状和方位。描述几何对象的数学函数依赖定义空间位置的坐标系。当地理参考系或坐标系变化时,几何对象是地理信息中唯一可能出现变化的项。如果空间是弹性连续的,地理对象的拓扑关系在变形时是不会发生变化的。比如当地理数据从一个坐标系变换为另一个时,在地理信息的上下文中,拓扑通常用来描述 n 维图的连通性,图的连续变换不会对其产生影响。计算拓扑学对几何图元连通性推导有更为详细的描述。

注 2: 电话号码被用于 11179-3 元模型的联系信息以及 11179-3 管理项合法性的时态范围中。

18.2 模块范围

本章描述了用于存储谁-什么-哪里-何时-那相当于那什么-怎样(W5H)事件数据的数据结构,即一个事件,通过谁(与事件关联的人/实体)、什么内容(主题/科目记录)、在何处(空间坐标数据)、在何时(时间数据)、是什么原因(外部关系),和怎样(语用细节)发生的进行描述。

18.3 模块功能

w5h_event 数据结构包含事件数据的要素。事件数据可能用来为描述事件提供语境。

示例 1: 下列是与“营业时间”相关的事件例子(0900~1700, 星期一~星期五), 例如, 与此事件相关的电话号码可以显示为“营业时间电话号码”:

```
business_hours_event: w5h_event_class =
(
( when_event = ( "time", 1, "0900/1700", (), (), (), ), ),
( when_event = ( "time", 1, "W1/W5", (), (), (), ), ),
( why_event = ( "characterstring", 0, "business hours", (), (), (), ), )
),
```

示例 2: 下列是与紧急事件“有关的事件例子, 如, 与此事件相关的电话号码可以显示为“紧急事件电话号码”:

```
emergency_event: w5h_event_class =
(
( why_event = ( "characterstring", 0, "emergency", (), (), (), ), ),
),
```

示例 3: 下面的例子是与“传真”相关的时间, 如, 将电话号码与传真相关联, 表示为“fax phone number”:

```
emergency_event: w5h_event_class =
(
( how_event = ( "characterstring", 0, "fax", (), (), (), ), ),
),
```

18.4 抽象模型

18.4.1 概述

18.4 的其余部分描述了 w5h_event_class 的对象模型。图 21 是 UML 对象模型。

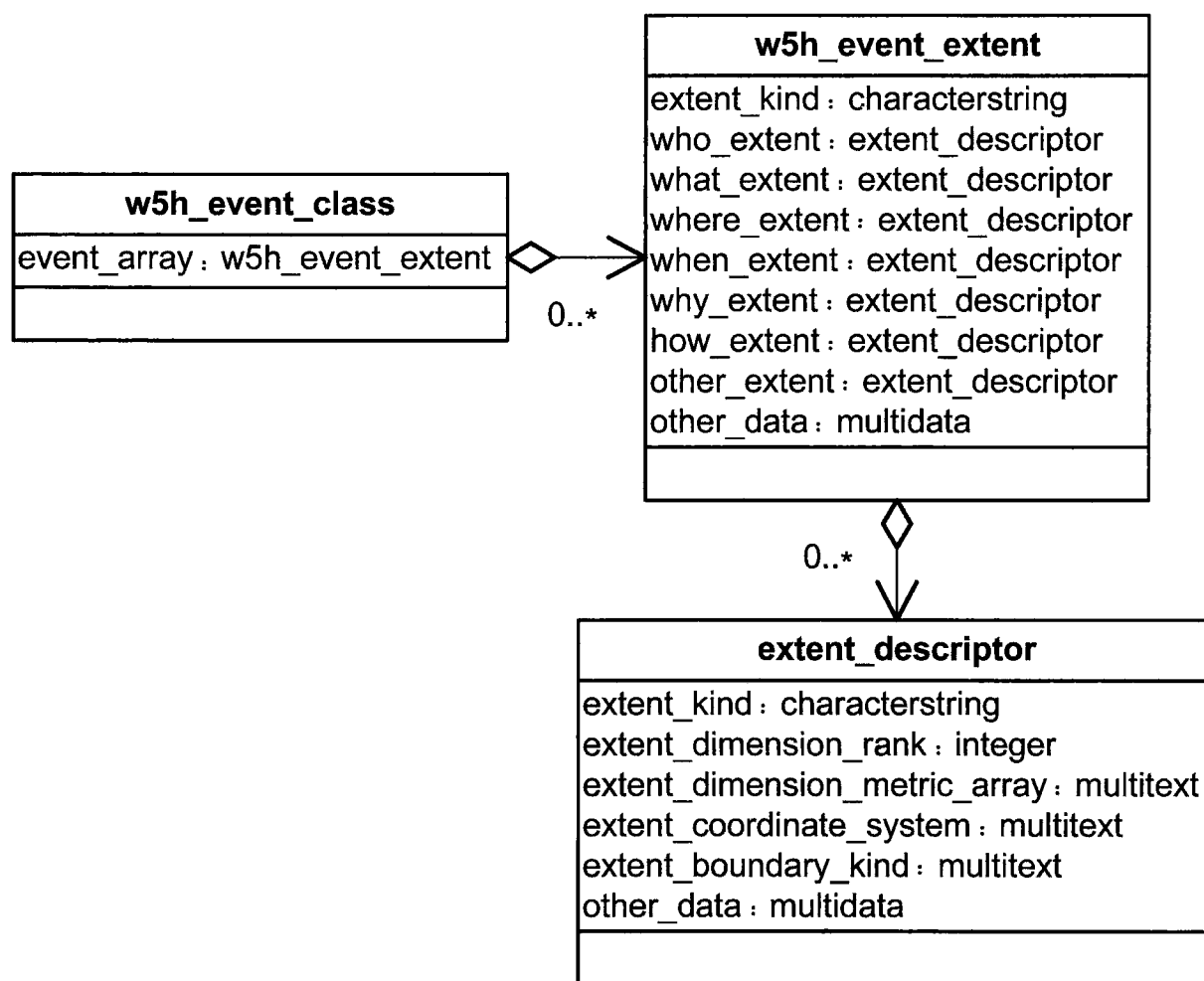


图 21 谁-什么-哪里-何时-原因-怎样(W5H)事件结构的 UML 表示

18.4.2 w5h_event_class

w5h_event_class 由一系列 W5H 事件范围组成。这个类包含如下的要素：

- event_array: array (0 .. *) of w5h_event_extent: 事件范围数组。

18.4.3 w5h_event_extent

w5h_event_extent 由一系列 W5H 事件范围组成。这个类别包括如下要素：

- event_kind: characterstring: 事件的类型，“范围”、“组”、“与算符”、“或算符”、“非算符”；
- who_extent: extent_descriptor: 与事件相关的实体-人-组 (EPG)，如人或与人有关的事件；
- 事件的种类，“范围”、“群”、“与算符”、“或算符”、“非算符”；
- what_extent: extent_descriptor: 与题目、主题、与/和类名有关的事件，如，分类学内的描述性本体或类名；
- where_extent: extent_descriptor: 空间范围与事件相关，如，纬度-经度的范围，或地理区域；

- when_extent: extent_descriptor: 时间与事件相关,如,日期范围,或从“石器时代”到“铜器时代”;
- why_extent: extent_descriptor: 外部构造与事件有关,如,工作流程、工序,或基本原理;
- how_extent: extent_descriptor: 与事件有关的语用细节;
- other_extent: extent_descriptor: 其他的范围记录与事件有关;
- other_data: multidata: 其他数据。

18.4.4 extent_descriptor

w5h_event_descriptor 保存了范围数据的集合。这个类包含了如下的要素:

- extent_kind: characterstring: 此要素属于种类范围;
- extent_dimension_rank: integer: 此要素属于范围中的维数,一列零显示一个量(0 维);
- extent_dimension_metric_array: multitext: 对于几何学来说,每个尺寸的度量单位,用括号分隔和围绕的空间,如三维的空间坐标“(度度底部)”以度(纬度)、度(经度)和高度(底部)作为各个尺寸的度量单位对于地志学来说,字符串“地志学”的已排序但未测量的各个尺寸,如二维布局体系“(地志学地志学)”;
- extent_dimension_coordinate_system: multitext: 每个用括号分隔和围绕的空间尺寸的坐标系,如“(纬度经度高度)”;
- extent_dimension_boundary_kind: multitext: 每个用括号分隔和围绕的空间尺寸的界限种类,如“(−180< =纬度<180−180< =经度<180 1000< =高度< = 60000)”;
- other_data: mutlidata: 其他与事件范围有关的数据。

18.5 计算的数据描述和数据类型

18.5.1 概述

18.5 使用 ISO/IEC 11404 符号定义了数据类型。

18.5.2 w5h_event_class

在 ISO/IEC 11404 中的定义:

```
type w5h_event_class = record
(
    event_array:
        array (0..*) of w5h_event_extent,
),
```

描述:

参见 18.4.2 中对记录和要素的描述。

18.5.3 w5h-事件-范围

在 ISO/IEC 11404 中的定义:

```
type w5h_event_extent = record
(
    event_kind:
        characterstring,
    who_extent:
```

```

        extent_descriptor,
    what_extent:
        extent_descriptor,
    where_extent:
        extent_descriptor,
    when_extent:
        extent_descriptor,
    why_extent:
        extent_descriptor,
    how_extent:
        extent_descriptor,
    other_extent:
        extent_descriptor,
    other_data:
        array (0..* ) of multidata,
),

```

描述:

参见 18.4.3 中对记录和要素的描述。

18.5.4 event_descriptor

在 ISO/IEC 11404 中的定义:

```
type event_descriptor = record
```

```

(
    extent_kind:
        characterstring,
    extent_dimension_rank:
        integer,
    extent_metric_array:
        multitext,
    extent_coordinate_system:
        multitext,
    extent_boundary_kind:
        multitext,
    other_data:
        multidata,
),

```

描述:

参见 18.4.4 中对记录和要素的描述。

18.6 绑定的附加条款

无绑定的附加条款。

18.7 一致性附加条款

无一致性附加条款。

19 模块 19:entity-person-group (EPG) 联系数据的数据结构

19.1 模块介绍

第 19 章描述了人、群、组织、或其他实体提供联系数据的数据结构。第 19 章将个人标识(第 16 章)、组织标识(第 16 章)、邮政数据(第 16 章)和电话号码(第 17 章)的数据结构与为详细联系数据提供使用和适用性的谁-什么-何处-何时-原因-怎样(W5H)事件数据(第 18 章)相结合。如人的联系数据可能包括在办公时间的电话号码(When 事件)、非办公时间的电话号码(不同的 When 事件)和紧急事件电话号码(Why 事件)。

19.2 模块范围

本章是为人、群、组织或其他实体提供接点数据的数据结构描述。

19.3 模块功能

contact_data 的数据结构包括联系数据的要素,如个人名称、组织名称、通讯地址、电话号码和使用的语境。

示例:下列是接点数据的例子,联系数据有 2 项:在法国的办公时间(0900~1700、星期一~星期五)的电话号码(+3311223344)和通讯地址;及紧急情况下联系方式(仅包含电话号码),电话号码(+3355667788):

```
person_x_contact_data: contact_data =
(
  ( // first piece of contact data
    ( // event data
      ( when_event = ( "time", 1, "0900/1700", (), (), (), ), ),
      ( when_event = ( "time", 1, "W1/W5", (), (), (), ), ),
      ( why_event = ( "characterstring", 0, "business hours", (), (), (), ), )
    ),
    ( // individual name
  ),
    ( // organization name
  ),
    ( // postal data
      (
        "int_upu_s42_10_05_addressee_form_of_address",
          "characterstring",
            "MONSIEUR",
        "int_upu_s42_10_06_addressee_given_name",
          "characterstring",
            "JEAN",
        "int_upu_s42_10_08_addressee_given_name",
          "characterstring",
            "DURAND",
        "int_upu_s42_14_24_delivery_point_location_street_number_or_plot",
          "characterstring",
```

```

        "25",
        "int_upu_s42_14_22_delivery_point_location_thoroughfare_type",
        "characterstring",
        "RUE",
        "int_upu_s42_14_21_delivery_point_location_thoroughfare_name",
        "characterstring",
        "DES FLEURS",
        "int_upu_s42_13_13_delivery_point_specification_postcode",
        "characterstring",
        "33500",
        "int_upu_s42_13_16_delivery_point_specification_town",
        "characterstring",
        "LIBOURNE",
        "int_upu_s42_13_14_delivery_point_specification_country",
        "characterstring",
        "FRANCE",
    ),
    ( // phone number
        ( ( contextualized_value_list =
            ( context_identifier = "ITU-T E.164",
              value = ( ( "ITU-T E.164", "iso-646", "+3311223344" ), ),
            ),
        ),
    ),
    ( // second piece of contact data
        ( // event data
            ( why_event = ( "characterstring", 0, "emergency", (), (), (), ), )
        ),
        ( // individual name
        ),
        ( // organization name
        ),
        ( // postal data
        ),
        ( // phone number
            ( ( contextualized_value_list =
                ( context_identifier = "ITU-T E.164",
                  value = ( ( "ITU-T E.164", "iso-646", "+3355667788" ), ),
                )
            ),
        ),
    ),
),
),

```

19.4 抽象模型

19.4.1 概述

19.4 的剩余部分描述了 contact_data_class 对象模型。图 22 是 UML 的对象模型。

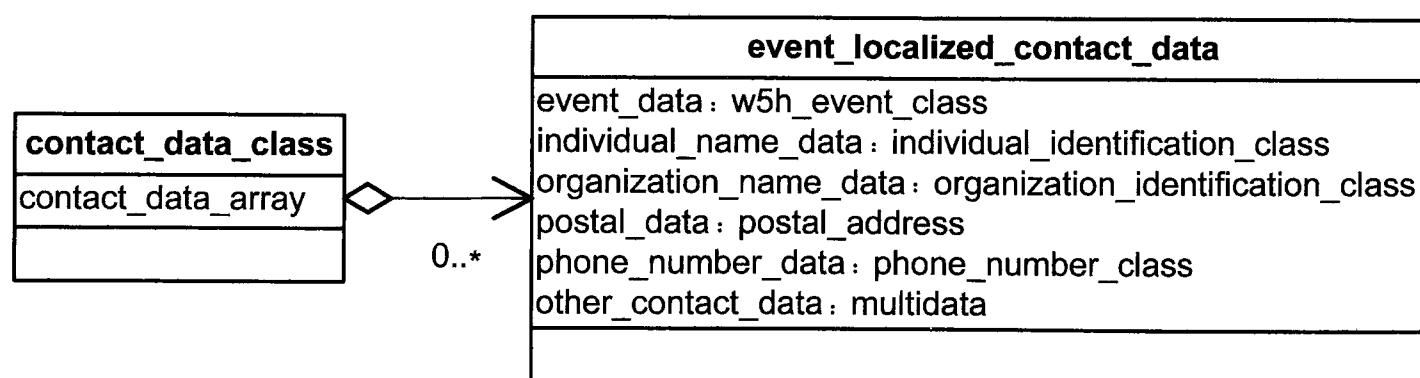


图 22 谁-什么-何处-为什么-怎么样(W5H)事件结构的 UML 表示

19.4.2 contact_data_class

contact_data_class 由一系列 event_localized_contact_data 结构组成。这个类别包括如下要素：

- contact_data_array: array (0 .. *) of event_localized_contact_data: 个人联系方式数据结构的数组。

19.4.3 event_localized_contact_data

event_localized_contact_data 由 W5H 事件范围和联系数据的要素组成。这个类别包括下列要素：

- event_data: w5h_event_class, 事件的类型；
- individual_name_data: array (0.. *) of individual_identification_class, 个人的姓名；
- organization_name_data: array (0.. *) of organization_identification_class, 组织的姓名；
- postal_data: array (0.. *) of postal_address, 实体的通讯地址；
- phone_number_data: array (0.. *) of phone_number_class, 实体的电话号码；
- other_contact_data: multidata, 其他的联系数据。本数据的结构和内容不在本条范围之内。

19.5 计算的数据描述和数据类型

19.5.1 概述

19.5 使用 ISO/IEC 11404 定义数据类型。

19.5.2 contact_data_class

在 ISO/IEC 11404 中的定义：

```

type contact_data_class = record
(
    contact_data_array:
        array (0.. *) of event_localized_contact_data,
),
  
```

描述：

参见 19.5.2 中对记录和要素的描述。

19.5.3 event_localized_contact_data

在 ISO/IEC 11404 中的定义：

```

type event_localized_contact_data = record
  
```

```
(
    event_data:
        w5h_event_class,
    individual_name_data:
        array (0..*) of individual_identification_class,
    organization_name_data:
        array (0..*) of organization_identification_class,
    postal_data:
        array (0..*) of postal_address,
    phone_number_data:
        array (0..*) of phone_number_class,
    other_contact_data:
        array (0..*) of multidata,
),
```

描述:

参见 19.4.3 中对记录和成分的数据描述。

19.6 绑定的附加条款

无绑定的附加条款。

19.7 一致性附加条款

无一致性附加条款。

20 模块 20:entity-person-group(EPG)安全性认证数据的数据结构

20.1 模块介绍

第 20 章是包括安全证书的一个容器数据结构。安全方法和技术在此国际标准中没有定义,但是在其他地方作了详细说明。

20.2 模块范围

本章描述了安全证书的数据结构。安全方法和技术在此国际标准中没有定义,但是规定了一个容器数据结构用来存储安全方法的证书和参数(如访问控制和认证机制)。

20.3 模块功能

security_credentials_data 数据结构包括安全证书的元素。依据使用的语境不同,可能会使用多种类型的证书。

示例:下列是联系方式数据的例子,联系方式数据有 2 项:“办公时间”的数据项(0900~1700,星期一~星期五)描述了法国的电话号码(+3311223344)和邮政地址;同时,“紧急情况”的数据项描述了在紧急境况下使用的电话号码(+3355667788)。

```
person_x_credentials: security_credentials_data =
(
    ( // credentials for user at "mycompany.com"
        ( // event data; describes credentials for use at "mycompany.com"
```

```

    ( where_event = ( "domain", 0, "mycompany.com", (), (), (), ), ),
),
( // individual credentials
  (
    "ssh-public", // kind
    "ssh-rsa", // method
    ( ( (), (), "public key for mycompany.com" ) ), // description
    ( // public key: value as multidata
      ( ( (),
        data = ( reflit_kind = literal, literal_value =
          (
            literal_kind = "text",
            literal_encoding = "ascii",
            literal_bintext = text,
            literal_value_as_text = "\
              AAAB3NzaC1yc2EAAAABIwAAAIEAuD\
              bb6rByKZFxuvLRgtX9f3TR8GY5oT0\
              HKMQYdI9sdO/TbSf/+",
          ) ) ),
    ),
  ),
(
  "password", // kind
  "cleartext", // method
  ( ( (), (), "login password for mycompany.com" ) ), // description
  ( // public key: value as multidata
    ( ( (),
      data = ( reflit_kind = literal, literal_value =
        (
          literal_kind = "text",
          literal_encoding = "ascii",
          literal_bintext = text,
          literal_value_as_text = "swordfish",
        ) ) ),
    ),
  ),
),
),
),

```

20.4 概念模型和对象模型

20.4.1 概述

20.4 的剩余部分描述了 security_credentials_data 类及其相关类的对象模型。这个对象模型对应于第 20.6 中绑定-独立的语法。图 23 是 UML 对象模型。

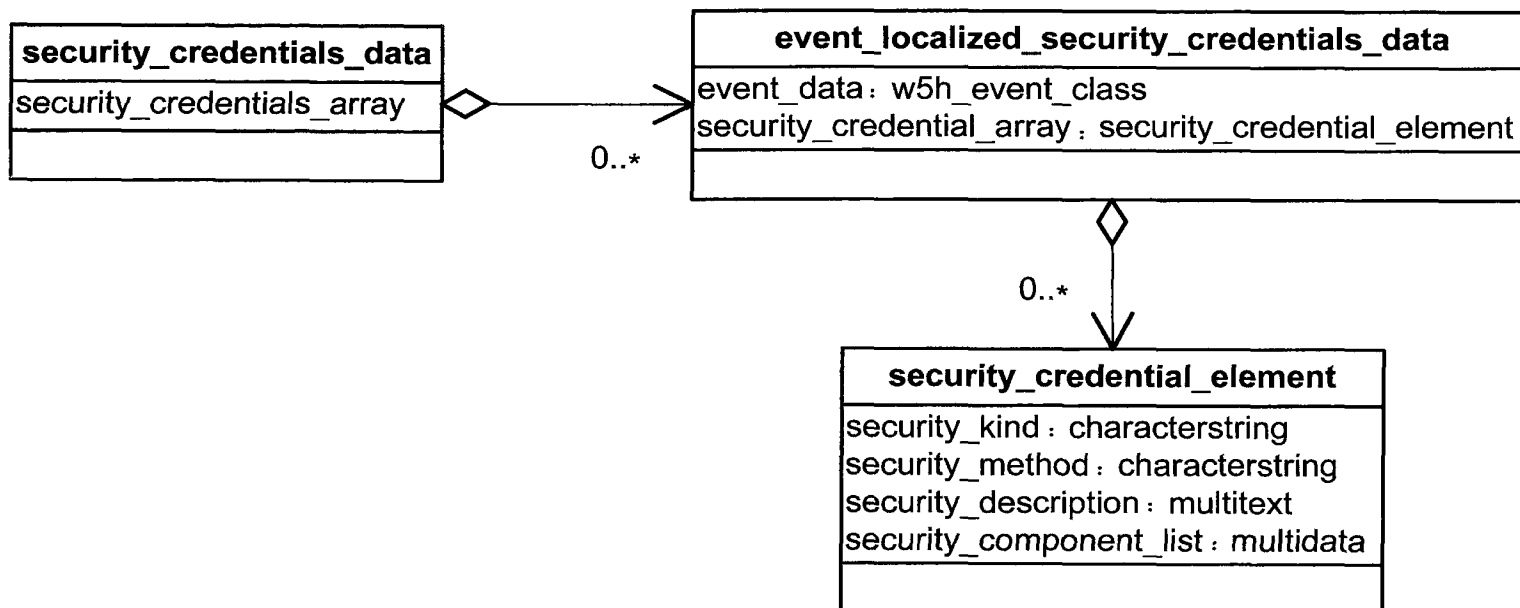


图 23 安全证书数据的 UML 表示

20.4.2 security_credentials_data

security_credentials_data 由一系列 W5H 事件范围组成。这个类别包括下列成员属性：
 ——security_credentials_array: array (0 .. *) event_localized_security_credentials_array,安全证书范围数组。

20.4.3 event_localized_security_credentials_data

event_localized_security_credentials_data 由 W5H 事件数据组成,本地化安全证书数组。这个类别包括下列成员属性：

- event_data: w5h_event_class,事件类型；
- security_credential_array: array (0.. *) of security_credential_element,安全证书；
- other_data: multidata,其他数据。

20.4.4 security_credential_element

security_credential_element 保存一个独立的安全证书系列。这个类别包括下列成员属性：

- security_kind: characterstring,此要素是证书类型；
- security_method: characterstring,此证书的安全方法；
- security_description: multitext,对安全类型、方法,和证书的描述；
- security_component_list: multidata,证书本身的数据；
- other_data: mutlidata,其他数据。

20.5 计算的数据描述和数据类型

本条使用 ISO/IEC 11404 符号定义了数据类型。

20.5.1 安全-证书-数据

在 ISO/IEC 11404 中的定义：

```

type security_credentials_data = record
(
    security_credentials_data_array:
    
```

```

        array (0..*) of event_localized_security_credentials_data,
    ),

```

描述:

参见 20.4.2 中对记录和要素的数据描述。

20.5.2 事件-本地化-安全-证书-数据

在 ISO/IEC 11404 中的定义:

```

type event_localized_security_credentials_data = record
(
    event_data:
        w5h_event_class,
    security_credential_array:
        array (0..*) of security_credential_element,
),

```

描述:

参见 20.4.3 中对记录和要素的数据描述。

20.5.3 安全-证书-元素

在 ISO/IEC 11404 中的定义:

```

type security_credential_element = record
(
    security_kind:
        characterstring,
    security_method:
        characterstring,
    security_description:
        multitext,
    security_component_list:
        multidata,
),

```

描述:

参见 20.4.4 中对记录和要素的数据描述。

20.6 绑定的附加条款

无绑定的附加条款。

20.7 一致性附加条款

无一致性附加条款。

21 模块 21: entity-person-group(EPG)关系和分组数据的数据结构

21.1 模块介绍

第 21 章描述了与 EPG 关系的容器数据结构。一个 EPG(实体-个人-群组)可能是一个实体(如一

个组织),一个个人(如个体本身)或者一个群组(如人与/或实体的组合)。这些数据结构为 EPGs 之间的关系提供表示方法。

21.2 模块范围

描述了与 EPG 关系的容器数据结构。一个 EPG(实体-个人-群组)可能是一个实体(如一个组织),一个个人(如个体本身)或者一个群组(如人与/或实体的组合)。这些数据结构为 EPGs 之间的关系提供表示方法。

21.3 模块功能

epg_relationship_data 数据结构包括实体、个人、与/或群组关系的元素。

示例:Bob 和 Carol(双亲)和 Ted 和 Alice(两个孩子)形成一个组群。他们的组群图可以表示为:

```
family: epg_relationship_data =
(
  "family", "bob_carol_ted_alice",
  (
    ("edge", ("parent" "bob"), "to", (), ()),
    ("edge", ("parent" "carol"), "to", (), ()),
    ("edge", ("child" "ted"), "to", (), ()),
    ("edge", ("child" "alice"), "to", (), ()),
  ),
  ()
)
```

21.4 概念模型和对象模型

21.4.1 概述

21.4 的剩余部分描述了 epg_relationship_data 类和其相关类。图 24 是 UML 对象模型。

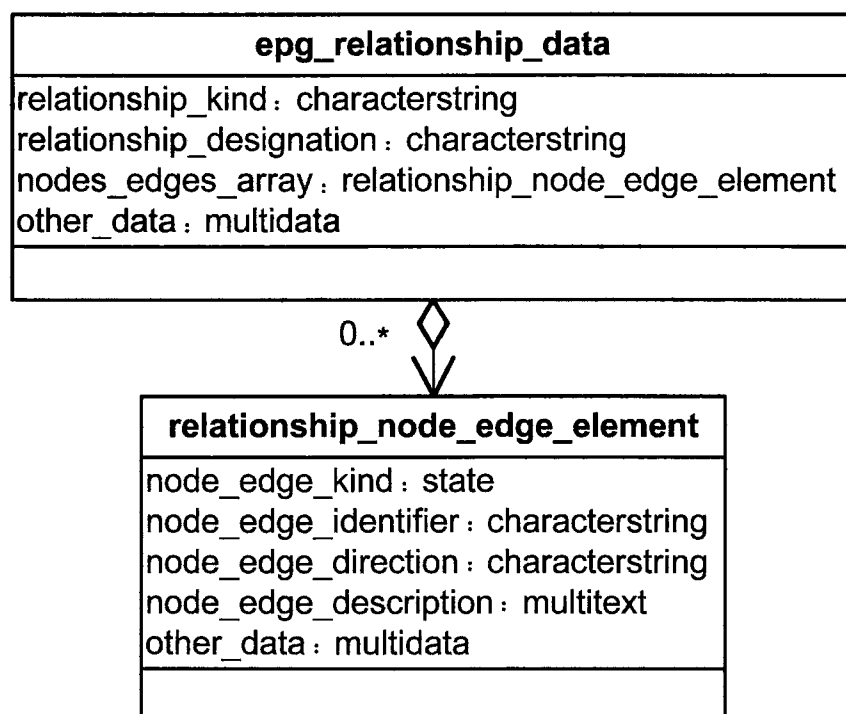


图 24 EPG 关系数据类的 UML 表示

21.4.2 epg_relationship_data

epg_relationship_data 类由下列要素组成:

- relationship_kind: characterstring, 关系类型, 如, “父母-孩子”;
- relationship_identifier: array (0..*) characterstring, 标识符与此关系相关;
- nodes_edges_array: array (0..*) of relationship_node_edge_element, 包含结点、边缘, 或两者的数组;
- other_data: array (0..*) of multidata, 其他数据。

21.4.3 relationship_node_edge_element

relationship_node_edge_element 由一系列 W5H 事件范围组成。这个类包含如下要素:

- node_edge_kind: state of (node, edge, unspecified), 元素的类型: 节点或边;
- node_edge_identifier: array (0..*) of characterstring, 与节点/边相关的标识符;
- node_edge_direction: array (0..*) of characterstring, 节点的方向(指向/指出/双向/无方向), 或边的方向;
- node_edge_description: multitext, 与节点/边相关的描述;
- other_data: multidata, 其他数据。

21.5 计算的数据描述和数据类型

21.5.1 概述

21.5 使用 ISO/IEC 11404 符号定义了数据类型。

21.5.2 epg_relationship_data

在 ISO/IEC 11404 中的定义:

```
type epg_relationship_data = record
(
    relationship_kind:
        characterstring,
    relationship_identifier:
        characterstring,
    nodes_edges_array:
        array (0..*) of relationship_node_edge_element,
    other_data:
        array (0..*) of multidata,
),
```

描述:

参见 20.5.2 中对记录和要素的数据描述。

21.5.3 关系-节点-边缘-元素

在 ISO/IEC 11404 中的定义:

```
type relationship_node_edge = record
(
```

node_edge_kind:
state of (node, edge, unspecified),
node_edge_identifier:
array (0..*) of characterstring,
node_edge_direction:
array (0..*) of characterstring,
node_edge_description:
multitext,
other_data:
array (0..*) of multidata,

),

描述:

参见 20.5.3 中对记录和要素的数据描述。

21.6 绑定的附加条款

无绑定的附加条款。

21.7 一致性附加条款

无一致性附加条款。

参 考 文 献

- [1] ISO/IEC 1087-1:2000, Terminology work—Vocabulary—Part 1: Theory and application
 - [2] ISO/IEC 2382-4:1999, Information technology—Vocabulary—Part 4: Organization of data
 - [3] ISO/IEC 2382-5:1999, Information technology—Vocabulary—Part 5: Representation of data
 - [4] ISO/IEC 2382-7:2000, Information technology—Vocabulary—Part 7: Computer programming
 - [5] ISO/IEC 2382-8, Information technology—Vocabulary—Part 8: Security
 - [6] ISO/IEC 2382-23:1994, Information technology—Vocabulary—Part 23: Text processing
 - [7] ISO/IEC 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability
 - [8] ISO/IEC 11179 (all parts), Information technology—Metadata registries (MDR)
 - [9] ISO/IEC 19763 (all parts), Information technology—Metamodel framework for interoperability (MFI)
 - [10] ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards, sixth edition, 2011
-

中 华 人 民 共 和 国
国 家 标 准
信 息 技 术 元 数 据 注 册 系 统 (MDR)
模 块

GB/T 30881—2014/ISO/IEC 19773:2011

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)

网址 www.spc.net.cn

总编室:(010)64275323 发行中心:(010)51780235

读者服务部:(010)68523946

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

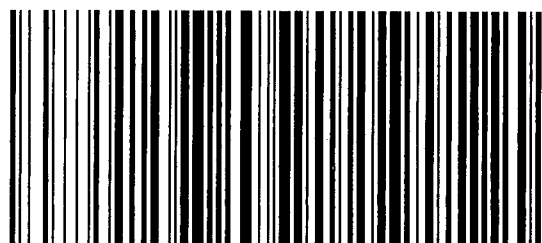
*

开本 880×1230 1/16 印张 5.25 字数 149 千字
2014年10月第一版 2014年12月第二次印刷

*

书号:155066·1-49608 定价 69.00 元

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68510107



GB/T 30881-2014