



中华人民共和国国家标准

GB/T 22022—2008/ISO 19108:2002

地理信息 时间模式

Geographic information—Temporal schema

(ISO 19108:2002, IDT)

2008-06-20 发布

2008-12-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言	III
引言	IV
1 范围	1
2 一致性	1
2.1 一致性类和要求	1
2.2 数据传输应用模式	1
2.3 带操作的数据应用模式	1
2.4 要素目录	1
2.5 元数据元素规范	1
2.6 数据集元数据	1
3 规范性引用文件	1
4 术语、定义和缩略语	2
4.1 术语和定义	2
4.2 缩略语	5
5 地理信息时间概念模式	5
5.1 地理信息时间概念模式结构	5
5.2 时间几何	6
5.3 时间参照系	14
5.4 时间位置	18
5.5 时间和地理信息组分	20
附录 A (规范性附录) 抽象测试套件	26
附录 B (资料性附录) 应用模式中时间的使用	27
附录 C (规范性附录) 在元数据中描述时间参照系	32
附录 D (资料性附录) 历法描述	35
附录 E (资料性附录) 本标准中的类名中英文对照表	43
参考文献	44

前 言

本标准等同采用 ISO 19108:2002《地理信息 时间模式》(英文版)。

为便于使用,本标准作了下列编辑性修改:

- a) 本标准的编写格式执行 GB/T 1.1—2000《标准化工作导则 第 1 部分:标准的结构和编写规则》的要求。
- b) 将“本国际标准”一词改为“本标准”。
- c) 删除了原国际标准的前言。
- d) 为适应我国国情,在资料性附录 D 中增加了条文 D.3.2。
- e) 为便于理解,增加了资料性附录 E。
- f) 5.2.3.5 中增加了注释;5.2.3.7 示例中增加了注释“译者注: $7 \times 10^{-3} \text{s}$ ”。
- g) 由于 ISO 19107、ISO 19111 和 ISO 19115 国际标准已经出版,在本标准规范性引用文件中删去了原国际标准中标识即将出版的角标。

本标准的附录 A 和附录 C 是规范性附录,附录 B、附录 D 和附录 E 是资料性附录。

本标准由全国地理信息标准化技术委员会提出。

本标准由全国地理信息标准化技术委员会归口。

本标准起草单位:中国测绘科学研究院、国家基础地理信息中心、中国地质调查局发展研究中心。

本标准主要起草人:苏山舞、姜作勤、李莉、殷红梅。

引 言

本标准定义描述从现实世界中抽象出来的地理信息的时间特性所需要的标准概念。地理信息的时间特性包括要素属性、要素操作、要素关联以及在时间域中取值的元数据元素。

计算机和地理信息系统的广泛应用增强了多学科地理空间数据的分析能力。地理空间数据不再限定在三维空间域,许多地理信息系统需要时间特性的数据。时间特性的标准概念模式将增强地理信息适应特定类型应用的能力,如:仿真和预测模拟。

作为一个基本的自然存在,时间与所有科学技术领域有关。本标准所描述的许多概念也适用于地理信息之外的领域。国际标准化组织地理信息标准化技术委员会(ISO/TC 211)无意制定独立的描述时间标准,但 ISO/TC 211 认为需要制定一个描述地理数据集和要素时间特性的标准。地理信息系统和地理信息软件开发者及用户将使用这个模式,提供一致的可理解的时间数据结构。

过去,要素的时间特性已经被作为专题要素的属性。例如,要素“建筑物”可以有属性“建筑日期”。然而,将要素的行为作为时间函数进行描述的需求越来越大。当将时间独立于空间处理时,这种需求可以得到有限程度的支持。例如,一个移动目标的运动路径可以用一个称为“路径点”的要素集来描述,被当作点描述的每一个路径点有一个属性,该属性提供该目标在相应空间位置的时间。如果时间维与空间维相结合,时间行为的描述可能更容易,因此,一个要素可以用一个时空对象来表示。例如:一个移动目标的路径可以表示为用一条 x 、 y 、 t 坐标描述的曲线。本标准是为使要素的时间属性的应用标准化而制定的。在本标准中虽然没有采用空间和时间坐标的组合来描述要素几何特性,但是已经为在 ISO 19100 系列中制定相应标准奠定了基础。

地理信息 时间模式

1 范围

本标准定义了描述地理信息时间特性的概念。本标准以现有的时间信息交换的信息技术标准为依据,提供用于定义时间要素属性、要素操作、要素关联和定义地理信息时间特性的元数据的基础。本标准主要考虑从现实世界抽象出来的地理信息时间特性,它强调有效时间,而不是事务处理时间。

2 一致性

2.1 一致性类和要求

本标准根据测试项的特点定义 5 种类别一致性。

2.2 数据传输应用模式

为与本标准保持一致,数据传输应用模式应满足附录 A 中 A.1 的要求。

2.3 带操作的数据应用模式

为与本标准保持一致,支持数据操作的应用模式应满足附录 A 中 A.2 的要求。

2.4 要素目录

为与本标准保持一致,要素目录应满足附录 A 中 A.3 的要求。

2.5 元数据元素规范

为与本标准保持一致,元数据规范应满足附录 A 中 A.4 的要求。

2.6 数据集元数据

为与本标准保持一致,数据集元数据应满足附录 A 中 A.5 的要求。

3 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准,然而,鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本标准。

GB/T 18221—2000 信息技术 程序设计语言、环境与系统软件接口 独立于语言的数据类型 (idt ISO/IEC 11404:1996)

GB 3102.1—1993 空间和时间的量和单位 (eqv ISO 31-1:1992)

GB 3100—1993 国际单位制及其应用 (eqv ISO 1000:1992)

GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法 (ISO 8601:2000, IDT)

GB/T 19710—2005 地理信息 元数据 (ISO 19115:2003, MOD)

ISO/TS 19103 地理信息 概念模式语言 (Geographic information—Conceptual schema language)

ISO 19107 地理信息 空间模式 (Geographic information—Spatial schema)

ISO 19109 地理信息 应用模式规则 (Geographic information—Rules for application schema)

ISO 19110 地理信息 要素编目方法 (Geographic information—Feature cataloguing methodology)

ISO 19111 地理信息 基于坐标的空间参照 (Geographic information—Spatial referencing by coordinates)

4 术语、定义和缩略语

4.1 术语和定义

本标准采用以下术语与定义。

4.1.1

日历 calendar

不连续的时间参照系,为定义分辨率为一日的时间位置提供基础。

4.1.2

日历时代 calendar era

从特定事件起算的,日历中使用的一类时段序列。

4.1.3

协调世界时 UTC; Coordinated Universal Time

由国际计量局与国际地球自转服务局(IERS)维护的时标,它构成协调发布标准频率和时间信号的基础。[ITU-R Rec. TF. 686-1 (1997)]

4.1.4

日 day

相当于地球自转周期的持续时间的时段。

4.1.5

边 edge

一维拓扑单形。[ISO 19107]

注:边的几何实现是曲线。在一个拓扑复形中一条边的边界是关联到该边的一个或两个节点的集合。

4.1.6

事件 event

在某时刻发生的活动。

4.1.7

要素 feature

现实世界现象的抽象。[ISO 19101]

注:要素可以作为类型或实例出现。具体应用时只能使用其中一种含义。

4.1.8

要素关联 feature association

连接一种要素类型的实例与相同或不同要素类型实例的关系。[ISO 19110]

注1:要素关联可以作为类型或实例出现,具体应用时只能使用其中一种含义。

注2:要素关联包括要素聚集。

4.1.9

要素属性 feature attribute

要素的特性。[ISO 19101]

注:要素属性有名称、数据类型和与之相关的值域等。

4.1.10

要素分割 feature division

一个先前存在的要素被两个或多个相同要素类型的不同要素实例取代时的要素更替。

示例:要素类型为“地块”的一个实例,在该地块被合法细分后,被更替为两个同类实例。

4.1.11

要素融合 feature fusion

两个或多个先前存在的属于同一要素类型的实例被另一个相同要素类型的实例所取代的要素更替。

示例：要素类型为“牧场”的两个实例，当牧场间的栅栏被移开时，被更替为一个同要素类型的实例。

4.1.12

要素操作 feature operation

要素类型的每个实例可以执行的操作。[ISO 19110]

示例：对“水坝”进行升高水坝的操作，其结果是该“水坝”的高度和“水库”水位的上升。

注：要素操作是提供要素类型定义的一种依据。

4.1.13

要素置换 feature substitution

一个要素实例被另一个相同或不同要素类型的要素实例所取代的一种要素更替。

示例：要素类型为“建筑物”的一个实例，在其被拆除后，被更替为要素类型为“停车场”的实例。

4.1.14

要素更替 feature succession

一个或多个要素实例被其他要素实例取代，被取代的要素实例不再存在。

4.1.15

几何单形 geometric primitive

表示空间上单一的、连通的、同质元素的几何对象。[ISO 19107]

注：几何单形是描述几何结构信息不能再分解的对象。它们包括点、曲线、面和体。

4.1.16

公历 Gregorian calendar

1582年引入的通用日历，所定义的年比儒略历的更接近回归年。[改编自ISO 8601:2000]

注：公历的引入消除了儒略年的累积误差。在公历中，历年或为平年，或为闰年，均分为12个顺序月。

4.1.17

时刻 instant

表示时间位置的0维几何单形。

注：时间几何在5.2阐述。

4.1.18

间隔标度 interval scale

具有任意原点的标度，它既用于描述值的顺序，又用于描述值之间的距离。

注：在间隔标度上度量值的比率没有意义。

4.1.19

儒略日期 Julian date

儒略日数加上从前一个平正午起已经消逝的该日的十进制小数。

4.1.20

儒略日数 Julian day number

约定儒略历从公元前4713年1月1日格林尼治平正午起算的时间消逝的日数。

4.1.21

生存期 life span

某事物存在的时段。

注：有效时间生存期是在现实模型中对象存在的时段。事务处理时间生存期是数据库对象在数据库中处在当前状态的时段。

4.1.22

月 month

约等于月亮朔望周期时间所持续的时段。

注：一个月持续时间是整日数。一个月中的日数由特定的日历规则确定。

4.1.23

结点 node

0 维拓扑单形。[ISO 19107]

注：结点的边界是空集。

4.1.24

顺序时代 ordinal era

按时间排序命名的时段集合中的一个时段。

4.1.25

顺序标度 ordinal scale

仅为度量对象相对位置提供基础的标度。

4.1.26

顺序时间参照系 ordinal temporal reference system

由顺序时代构成的时间参照系。

4.1.27

时段 period

表示时间范围的一维几何单形。

注：时段以两个不同的时间位置为边界。

4.1.28

时段时间 periodic time

一个周期的持续时间。[改编自 ISO 31-2:1992]

4.1.29

点 point

表示位置的 0 维几何单形。[ISO 19107]

注：点的边界是空集。

4.1.30

时间坐标 temporal coordinate

到作为时间坐标系基础的间隔标度原点的距离。

4.1.31

时间坐标系 temporal coordinate system

基于间隔标度的时间参照系，在间隔标度上的距离是以 1 个时间单位的倍数度量的。

4.1.32

时间要素关联 temporal feature association

与时间或时间约束有关的要素关联。

4.1.33

时间要素操作 temporal feature operation

以时间函数定义的要素操作。

4.1.34

时间位置 temporal position

相对于时间参照系的位置。

4.1.35

时间参照系 temporal reference system

度量时间的参照系。

4.1.36

拓扑复形 topological complex

在边界操作中闭合的拓扑单形的组合。[ISO 19107]

注：在边界操作中闭合，意味着如果一个拓扑单形在该拓扑复形中，则它的边界对象也在该拓扑复形中。

4.1.37

拓扑单形 topological primitive

单一的、不可分解的拓扑对象。[ISO 19107]

注：在几何实现中一个拓扑单形对应于一个同维的几何单形的内部。

4.1.38

事务处理时间 transaction time

事项在数据库中处于被处理并可能被查询的时间。[Jensen et al. (1994)]

4.1.39

有效时间 valid time

在抽象世界中，事项为“真”的时间。[Jensen et al. (1994)]

4.2 缩略语

下列缩略语适用于本标准。

AD 公元后 (Anno Domini)

BC 公元前 (Before Christ)

GPS 全球定位系统 (Global Positioning System)

TOW 周时间 (Time of Week)

UML 统一建模语言 (Unified Modelling Language)

UTC 协调世界时 (Coordinated Universal Time)

WN 周数 (Week Number)

5 地理信息时间概念模式

5.1 地理信息时间概念模式结构

本条提出描述地理信息时间特性的概念模式，它是用统一建模语言 (UML) 定义的 [Object Management Group (1999)]。ISO/TS 19103 描述了在 ISO 19100 系列标准中使用 UML 的方法。一个 UML 类的三项基本内容是属性、操作和关联，本模式使用了全部三者。本模式是一个抽象模型。为与本标准一致，实现应提供由本抽象模型所描述的能力，但是并不需要以同样方式实现。

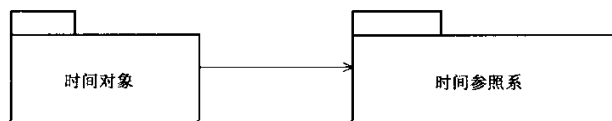


图 1 时间模式的结构

模式由两个包构成 (图 1)。时间对象包 (见 5.2) 定义时间的几何和拓扑对象，应被用作要素和数据集时间特性的值。对象的时间位置应相对于时间参照系确定。时间参照系包 (5.3, 5.4) 提供描述时间参照系的元素。5.5 阐述 5.2~5.4 描述的概念如何在地理信息的相关环境中应用。

在 ISO 19100 系列标准中定义的 UML 类的名称都冠以前缀，它以两个字符开头并紧接一个下划线，以便标识按这些类定义的特定标准或包。TM_用于标识本标准定义的类。

5.2 时间几何

5.2.1 时间维

时间是一个类似于空间维的维。像空间一样,时间也有几何特征和拓扑特征。时间的一个点占据一个位置,这个位置可以基于时间坐标系确定。时间的距离可以度量。但是,与空间不同,时间具有单一维。时间参照系类似于一些应用中用于描述空间位置的线性参照系。虽然时间具有一个绝对的方向,即时间的运动总是向前,但是时间却可以从两个方向度量。

注:虽然在概念层面上时间总是具有几何特征和拓扑特征,但有时可能或需要仅描述几何特征或仅描述拓扑特征。

度量时间可采用顺序标度和间隔标度两种类型的标度。顺序标度仅提供关于时间相对位置的信息,间隔标度提供度量持续时间的基础。

5.2.2 时间对象

时间的几何和拓扑对象应被用作时间要素和数据集时间特性的值。解释和示例见 5.5 和附录 B。TM_Object(TM_对象)(图 2)是一个抽象类,它有两个子类。TM_Primitive(TM_单形)是抽象类,它表示不可再分的时间几何或拓扑元素。TM_Primitive 有两个子类。TM_GeometricPrimitive(TM_几何单形)(5.2.3)提供关于时间位置的信息。TM_TopologicalPrimitive(TM_拓扑单形)(5.2.4.2)提供关于时间连通性的信息。TM_Complex(TM_复形)是 TM_Primitive 的聚集。TM_TopologicalComplex(TM_拓扑复形)(5.2.4.5)是在本标准中定义的 TM_Complex 的唯一子类,它是相连的 TM_TopologicalPrimitive 的聚集。

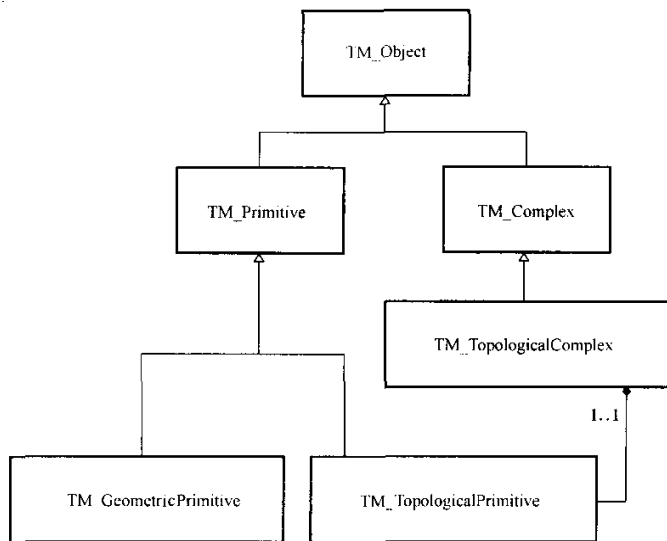


图 2 时间对象

5.2.3 时间几何单形

5.2.3.1 时间几何单形类

时间维有两个几何单形,分别是时刻和时段。其定义与度量时间的方法有关,当用时间的间隔标度度量时间时,采用解析方法定义;当用时间的顺序标度度量时间时,采用类推法定义。TM_GeometricPrimitive 是一个有两个子类 TM_Instant(TM_时刻)和 TM_Period(TM_时段)的抽象类(图 3)。TM_GeometricPrimitive 从 TM_Primitive 继承了 TM_Order(TM_顺序)接口上的依赖关系,并且有一个在 TM_Separation(TM_间隔)接口上的依赖关系。依赖关系的 <<uses>> 构造型意指该类可以支持为接口定义的任一操作,但不需要支持全部操作。

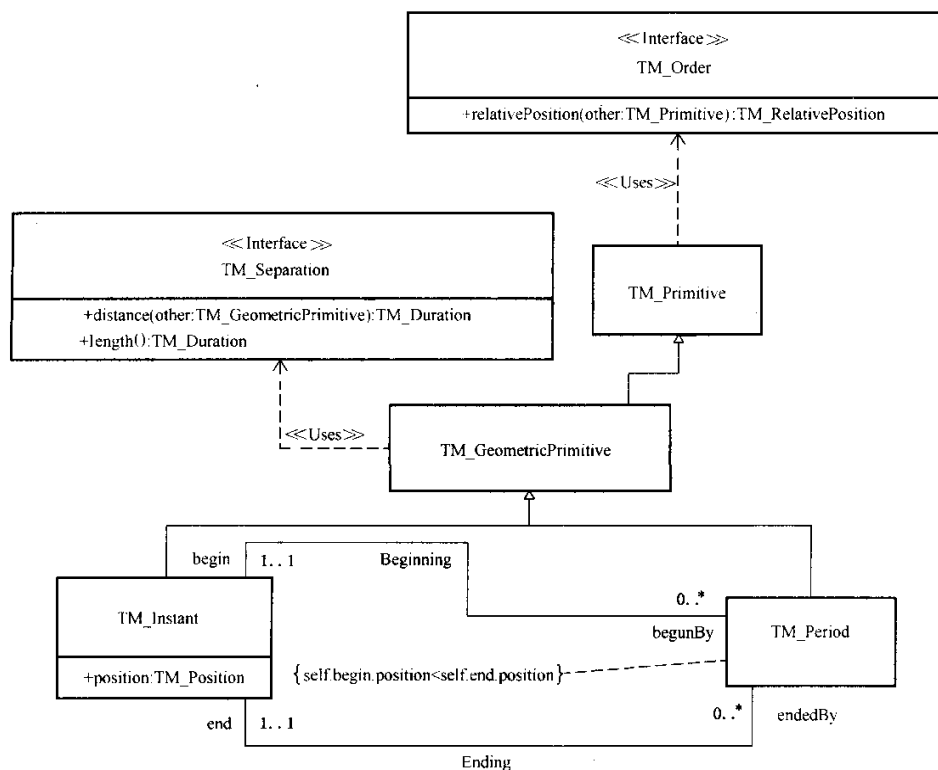


图 3 时间几何单形

5.2.3.2 TM_Instabt

时刻是表示时间位置的 0 维几何单形。它相当于空间的点。实际上,时刻是持续时间小于时间标度分辨率的时间间隔。

属性:

TM_Instabt 有一个属性。

- a) position: TM_TemporalPosition(TM_时间位置)应提供该 TM_Instabt 的位置。TM_TemporalPosition 应与一个时间参照系关联,详见 5.3。TM_Instabt 的实例是可标识的对象,而 TM_TemporalPosition 的实例是数据值。一个给定的 TM_Instabt 的 TM_Position(TM_位置)可以被与不同时间参照系关联的一个等效 TM_TemporalPosition 更替。

5.2.3.3 TM_Period

时段是表示时间长度的一维几何单形。时段等效于空间的一条曲线。像曲线一样,它是由起点和终点(时刻)为界的开区间并且具有长度(持续时间)。在时间上它的位置由起点和终点时刻的时间位置描述,它的长度等于这两个时间位置之间的时间距离。

由于在一个顺序标度上持续时间是无法度量的,因此无法在时段中区分时刻。实际上以顺序标度量度时间时,单一事件发生的时间可以看作时刻。一系列顺序事件一定占据一个时间间隔,即时段。“时段”通常用于区分具有共同特性的事件的顺序。

关联:

- a) Beginning 将 TM_Period 连接到开始时的 TM_Instant。
- b) Ending 将 TM_Period 连接到结束时的 TM_Instant。

由于多种原因,以起点或终点规定的 TM_Instant 的位置可能是不确定的。有关不确定的时间位置的讨论见 5.4.3。

约束:

- a) {self. begin. position < self. end. position} 说明时段的开始时刻位置小于(即早于)该时段结束的时间位置。

5.2.3.4 TM_Order

TM_GeometricPrimitive 从 TM_Primitive 继承了 TM_Order 上的依赖关系, TM_Order 提供确定这个 TM_Primitive 相对于另一个 TM_Primitive 位置的操作。

操作:

- a) relativePosition(other; TM_Primitive): TM_RelativePosition 应接受另一个 TM_Primitive 作为输入并按 5.2.3.5 的规定返回 TM_RelativePosition(TM_相对位置)值。

5.2.3.5 TM_RelativePosition

枚举数据类型的 TM_RelativePosition(图 4)提供相对位置的值,这些值基于 Allen(1983)确定的 13 种时间关系。对于 TM_Primitive,操作 TM_Order.relativePosition 将返回一个 TM_RelativePosition 值,如下:

- a) 如果两个 TM_Primitive 都是 TM_Instant,该操作应返回如下 TM_RelativePosition 值:

返回值:	条件:	注释:
Before	self. position < other. position	该时刻早于另一时刻
Equals	self. position = other. position	该时刻等于另一时刻
After	self. position > other. position	该时刻晚于另一时刻
- b) 如果该 TM_Primitive 是 TM_Period,而另一 TM_Primitive 是 TM_Instant,该操作应返回如下 TM_RelativePosition 值:

返回值:	条件:	注释:
Before	self. end. position < other. position	该时段结束时刻早于另一时刻
EndedBy	self. end. position = other. position	该时段结束时刻等于另一时刻
Contains	self. begin. position < other. position	该时段开始时刻早于另一时刻并且
	AND self. end > other. position	该时段结束时刻晚于另一时刻
BegunBy	self. begin. position = other. position	该时段开始时刻等于另一时刻
After	self. begin. position > other. position	该时段开始时刻晚于另一时刻
- c) 如果该 TM_Primitive 是 TM_Instant 而另一 TM_Primitive 是 TM_Period,该操作应返回如下 TM_RelativePosition 值:

返回值:	条件:	注释:
Before	self. position < other. begin. position	该时刻早于另一时段开始时刻
Begins	self. position = other. begin. position	该时刻等于另一时段开始时刻
During	self. position > other. begin. position	该时刻在另一时段开始时刻与结束时刻之间
	AND self. position < other. end. position	
Ends	self. position = other. end. position	该时刻等于另一时段结束时刻
After	self. position > other. end. position	该时刻晚于另一时段结束时刻

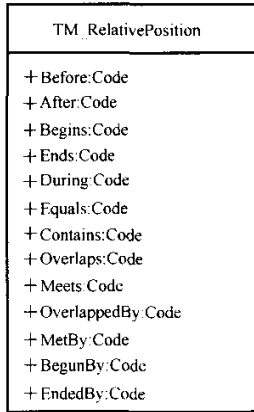


图 4 TM_RelativePosition

d) 如果两个 TM_Primitive 都是 TM_Period, 这个操作应返回如下 TM_RelativePosition 值:

返回值:	条件:	注释:
Before	self. end. position < other. begin. position	该时段结束时刻早于另一时段开始时刻
Meets	self. end. position = other. begin. position	该时段结束时刻等于另一时段开始时刻
Overlaps	self. begin. position < other. begin. position AND self. end. position > other. begin. position AND self. end. position < other. end. position	该时段开始时刻早于另一时段开始时刻, 并且该时段结束时刻在另一时段开始与结束时刻之间
Begins	self. begin. position = other. begin. position AND self. end. position < other. end. position	该时段开始时刻等于另一时段开始时刻, 并且该时段结束时刻早于另一时段结束时刻
BegunBy	self. begin. position = other. begin. position AND self. end. position > other. end. position	该时段开始时刻等于另一时段开始时刻, 并且该时段结束时刻晚于另一时段结束时刻
During	self. begin. position > other. begin. position AND self. end. position < other. end. position	该时段开始时刻晚于另一时段开始时刻, 并且该时段结束时刻早于另一时段结束时刻
Contains	self. begin. position < other. begin. position AND self. end. position > other. end. position	该时段开始时刻早于另一时段开始时刻, 并且该时段结束时刻晚于另一时段结束时刻
Equals	self. begin. position = other. begin. position AND self. end = other. end. position	该时段开始时刻等于另一时段开始时刻, 并且该时段结束时刻等于另一时段结束时刻

OverlappedBy	$self.begin.position > other.begin.position$ AND $self.begin.position < other.end.position$ AND $self.end.position > other.end.position$	该时段开始时刻在另一时段开始与结束时刻之间,并且该时段结束时刻晚于另一时段结束时刻
Ends	$self.begin.position > other.begin.position$ AND $self.end.position = other.end.position$	该时段开始时刻晚于另一时段开始时刻,并且该时段结束时刻等于另一时段结束时刻
EndedBy	$self.begin.position < other.begin.position$ AND $self.end.position = other.end.position$	该时段开始时刻早于另一时段开始时刻,并且该时段结束时刻等于另一时段结束时刻
MetBy	$self.begin.position = other.end.position$	该时段开始时刻等于另一时段结束时刻
After	$self.begin.position > other.end.position$	该时段开始时刻晚于另一时段结束时刻

如果任一 TM_Position 输入值是不确定的,操作将导致例外。

5.2.3.6 TM_Separation

TM_GeometricPrimitive 有一个在接口 TM_Separation 上的依赖关系,它提供计算长度和距离的操作。TM_Duration(TM_持续时间)(图 5)是一个包括那些操作返回值的数据类型。

- a) length(): TM_Duration 应返回该 TM_GeometricPrimitive 的持续时间。TM_Instant 的长度定义为 0。当 TM_GeometricPrimitive 是 TM_Period 时,该操作应返回由 TM_Period.begin 和 TM_Period.end 确定的时间位置之间的距离。如果任一 TM_TemporalPosition 值是不确定的或者 TM_TemporalPosition 与一个 TM_OrdinalReferenceSystem(TM_顺序参照系)相关联,操作将导致例外。
- b) distance(other: TM_GeometricPrimitive): TM_Duration 应返回从一个 TM_GeometricPrimitive 到另一个 TM_GeometricPrimitive 的距离,例如:它们时间的位置差的绝对值。距离是两个 TM_GeometricPrimitive 的两个 TM_TemporalPosition 之间最近的距离。如果任一 TM_GeometricPrimitive 与另一个连接、相交或被包含到另一个 TM_GeometricPrimitive 中,操作返回值为 0。下列情况操作将导致例外(1)两个 TM_Position 之任一个是不确定的,(2)两个 TM_Position 不是基于同一 TM_ReferenceSystem(TM_参照系),或(3)任一 TM_Position 是基于 TM_OrdinalReferenceSystem。

5.2.3.7 TM_Duration

TM_Duration(图 5)是用于描述时间维长度或距离的数据类型。它有两个子类型。

TM_PeriodDuration(TM_时段持续时间)使用由 GB/T 7408—2005 规定的时段持续时间的信息交换格式。它允许用多个时间单位,确切地说是用年、月、日、时、分、秒等表示持续时间。尽管个体值是可选的,但一个值至少应提供一个单位。

属性:

- a) designator: CharacterString = P 为必选元素,指明下列字符表示时段的持续时间。
- b) years[0..1]: CharacterString 为正整数,后接字符“Y”,指明时段的年数。
- c) months[0..1]: CharacterString 为正整数,后接字符“M”,指明时段的月数。
- d) days[0..1]: CharacterString 为正整数,后接字符“D”,指明时段的日数。
- e) timeIndicator[0..1]: CharacterString = “T”当时间序列包括小于一日的单位值时,则应包括该属性。

- f) hours[0..1]:CharacterString 为正整数,后接字符“H”,指明时段的小时数。
- g) minutes[0..1]:CharacterString 为正整数,后接字符“M”,指明时段的分钟数。
- h) seconds[0..1]:CharacterString 为正整数,后接字符“S”,指明时段的秒数。

最末位时间单位的值也可以采用非正整数的十进制小数值。

示例:持续时间 5 日 4 小时 30.7 分钟被表示为 P5DT4H30.7M。

注:尽管在 GB/T 7408—2005 中定义的这种格式用于公历的日期和 UTC 的时间,TM_PeriodDuration 也可以作为描述长度与距离的数据类型,只要时间位置是基于年、月、日描述日期的日历和用时、分、秒描述时间的时钟。

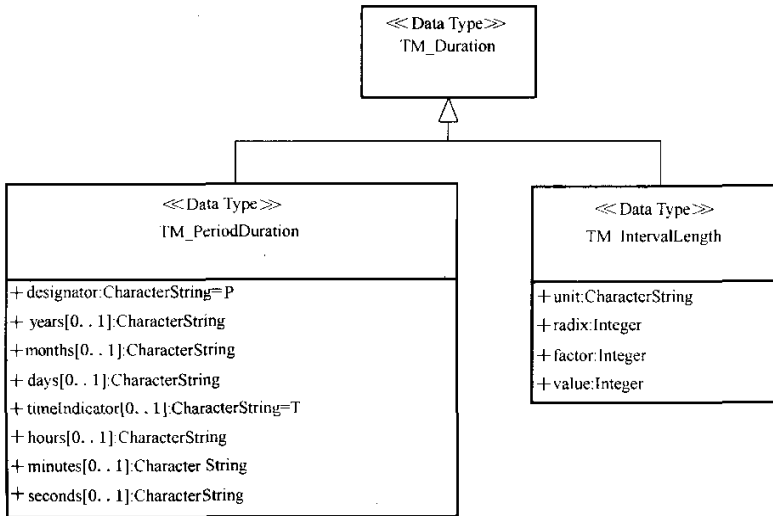


图 5 TM_Duration

此处 在 UML 中表示的 TM_IntervalLength(TM_间隔长度)是由 GB/T 18221—2000 规定用于表示时间间隔的数据类型。它用一个时间单位的倍数表示持续时间。

属性:

- a) unite:CharacterString 是表示间隔长度度量单位的名称。
- b) radix:Integer 是一个正整数,是单位因数的底数。
- c) factor:Integer 是一个整数,是底数的指数。
- d) value:Integer 是时间间隔的长度,是指定 $radix^{(-factor)}$ 的整倍数。

示例: unite=“秒”,radix=10,factor=3,value=7,表示 7 ms 的时间间隔长度(译者注: 7×10^{-3} s)。

5.2.4 时间拓扑对象

5.2.4.1 概述

拓扑提供有关对象之间在时间域中连通性的信息,有时也提供关于对象在时间域中的顺序信息。它不提供关于时间位置的信息。拓扑关系通常源于几何信息,然而时间位置的数据有时不能充分表示拓扑关系,因此,需要拓扑显式地表示。拓扑关系可以派生,但是拓扑仍可在要求显式描述拓扑关系的应用中使用。

示例:在一个顺序时代中,观测几个事件或状态的顺序是可能的,但是顺序时间参照系不支持为这些事件或状态赋予不同的时间位置。可以用拓扑单形建立这些事件或状态的模型来描述顺序。

5.2.4.2 TM_TopologicalPrimitive

拓扑单形表示单一的不可分割的拓扑元素以及与在拓扑复形中的其他拓扑单形的关系。与时间信

息相关的两类拓扑单形是 0 维的结点和 1 维的边。在本时间模式中, TM_TopologicalPrimitive (图 6) 有两个子类: TM_Node(TM_结点) 和 TM_Edge(TM_边)。当一个应用包括时间位置以及连通信息时, 一个 TM_TopologicalPrimitive 可以与一个同维的 TM_GeometricPrimitive 关联。由于拓扑单形用于提供连通信息, 它们最显著的特性是它们彼此连接的关联。另外要求每个 TM_TopologicalPrimitive 是且仅是一个 TM_TopologicalComplex 的成员。

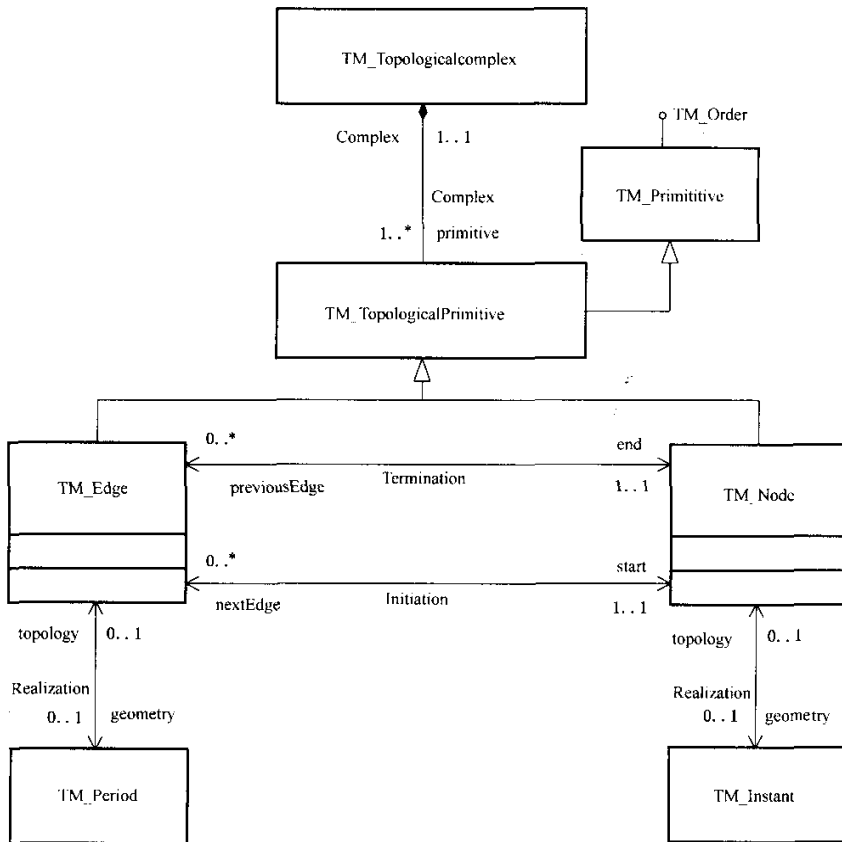


图 6 时间拓扑

5.2.4.3 TM_Node

TM_Node 是 0 维时间拓扑单形。它的几何实现是 TM_Instance。

关联:

TM_Node 应支持 3 种关联:

- Initiation 应连接作为 TM_Edge 起点的 TM_Node 到该边。
- Termination 应连接作为 TM_Edge 终点的 TM_Node 到该边。
- Realization 是一个可选关联, 它可连接 TM_Node 到与之相应的 TM_Instance。只有一个 TM_Node 可以同一个 TM_Instance 关联, 只有一个 TM_Instance 可以同一个 TM_Node 关联。

5.2.4.4 TM_Edge

TM_Edge 是一维时间拓扑单形。它对应于一个 TM_Period。

关联:

TM_Edge 应支持 3 种关联:

- a) Initiation 应连接 TM_Edge 到作为其起点的 TM_Node。一个 TM_Edge 可有且仅有一个起结点。
- b) Termination 应连接 TM_Edge 到作为其终点的 TM_Node。一个 TM_Edge 可有且仅有一个终结点。
- c) Realization 是一个可选关联,它应连接 TM_Edge 到与之相应的 TM_Period。只有一个 TM_Edge 可以与一个 TM_Period 关联,只有一个 TM_Period 可以与一个 TM_Edge 关联。

5.2.4.5 TM_TopologicalComplex

拓扑复形是一组相连的拓扑单形。某个拓扑单形总是与一个或多个其他的拓扑单形相连,因此,它总是拓扑复形的组分。在本模式中时间拓扑复形用 TM_TopologicalComplex 类表示。

关联:

- a) Composition 应将 TM_TopologicalComplex 与它所包含的 TM_TopologicalPrimitive 集相连。由于在 TM_TopologicalComplex 中每个 TM_Edge 连接两个 TM_Node,在复形中 TM_Node 的最小数是 2。

5.2.4.6 线性和非线性图

5.2.4.6.1 非线性图

在 Initiation 和 Termination 关联(图 6)的 TM_Edge 端点的多重性允许非线性拓扑。一个 TM_Node 可以是多于一个 TM_Edge 的 startNode 或 endNode。由于表示不同要素的时间特性或表示相同要素的不同时间特性,共享 startNode 或 endNode 的 TM_Edges 将以某种方式被分开。

注:时间非线性拓扑类似于空间的非平面情况。在这种情况下,以为相交或重叠的拓扑单形在不可测量的附加维上实际是分开的。

5.2.4.6.2 线性图

由于时间是单一维,时间拓扑应表示为线性图。在线性拓扑中,TM_TopologicalComplex 是 TM_Node 与 TM_Edge 交替排列的 TM_Primitive 的序列。这个序列的第一个元素是该序列中第一个 TM_Edge 的起结点,这个序列的最后一个元素是该序列最后一个 TM_Edge 的终结点。为了将应用模式限制为线性拓扑,Initiation 和 Termination 关联的 TM_Edge 端点多重性应被限制为 0..1,使得除第一个和最后一个 TM_Node 外,每个 TM_Node 应且仅连接两个 TM_Edge,一条是 previousEdge,另一条是 nextEdge。

5.2.4.7 TM_Order

TM_TopologicalPrimitive 从 TM_Primitive 继承了接口 TM_Order。TM_Order 提供一个用于定义该 TM_Primitive 相对于另一 TM_Primitive 位置的操作。

操作:

- a) RelativePosition (other:TM_Primitive):TM_RelativePosition 应接受一个 TM_Primitive 作为输入并返回以下规定的 TM_RelativePosition 值。

两个 TM_TopologicalPrimitive 之间的相对位置依赖于构成一个 TM_TopologicalComplex 的 TM_TopologicalPrimitive 在序列中占据的位置。对于 TM_TopologicalPrimitive,这个操作应返回一个枚举数据类型 TM_RelativePosition(图 4)的下列值:

返回值:	条件:
Before	在该序列中,该 TM_TopologicalPrimitive 在 other 之前,并且在 Initiation 或 Termination 关联上不被连接到 other。
Meets	两个 TM_TopologicalPrimitive 都是 TM_Edge,它们与同一个 TM_Node 相关联,并且在 Termination 关联中连接到该 TM_Node 的 TM_Edge 是 previousEdge,在 Initiation 关联中连接到该 TM_Node 的 other TM_Edge 是 nextEdge。

Begins	该 TM_TopologicalPrimitive 是 TM_Node, other TM_TopologicalPrimitive 是 TM_Edge, 两个 TM_Primitive 连接在一个 Initiation 关联中。
BegunBy	该 TM_TopologicalPrimitive 是 TM_Edge, other TM_TopologicalPrimitive 是 TM_Node, 两个 TM_Primitive 连接在一个 Initiation 关联中。
Equals	该 TM_TopologicalPrimitive 与 other 相同。
Ends	该 TM_TopologicalPrimitive 是 TM_Node, other TM_TopologicalPrimitive 是 TM_Edge, 两个 TM_Primitive 连接在一个 Termination 关联中。
EndedBy	该 TM_TopologicalPrimitive 是 TM_Edge, other TM_TopologicalPrimitive 是 TM_Node, 两个 TM_Primitive 连接在一个 Termination 关联中。
MetBy	两个 TM_TopologicalPrimitive 是连接到同一个 TM_Node 的 TM_Edge, 在 Initiation 关联中连接到 TM_Node 的该 TM_Edge 是 nextEdge, 在 Termination 关联连接到 TM_Node 的 other TM_Edge 是 previousEdge。
After	在该序列中, 该 TM_TopologicalPrimitive 在 other 之后, 并且在 Initiation 或 Termination 关联上不被连接到 other。

如果两个 TM_TopologicalPrimitive 不在同一个 TM_TopologicalComplex 之中, 操作将导致例外。

5.3 时间参照系

5.3.1 时间参照系类型

时间域中的值是相对于时间参照系度量的时间位置。GB/T 7408—2005 规定使用公历、当地时间和 UTC 进行信息交换, 这应是基于地理信息的主要时间参照系。对于一些地理信息应用, 需要不同的时间参照系。在这种情况下, 与一个应用模式或数据集相关的要素目录或元数据应包括对于描述时间参照系的文件的引用, 或者描述该时间参照系。当一个单独的要素目录、应用模式或数据集中使用了多于一个时间参照系时, 每一个时间特性的定义应标识被使用的时间参照系。本条描述应作为上述描述基础的概念模式。附录 C 定义应在上述描述中使用的元数据元素, 它们是由本模式派生的。

时间参照系包括三种公共时间参照系类型: 日历(在高分辨率时与时钟一起使用)、时间坐标系和顺序时间参照系(见图 7)。

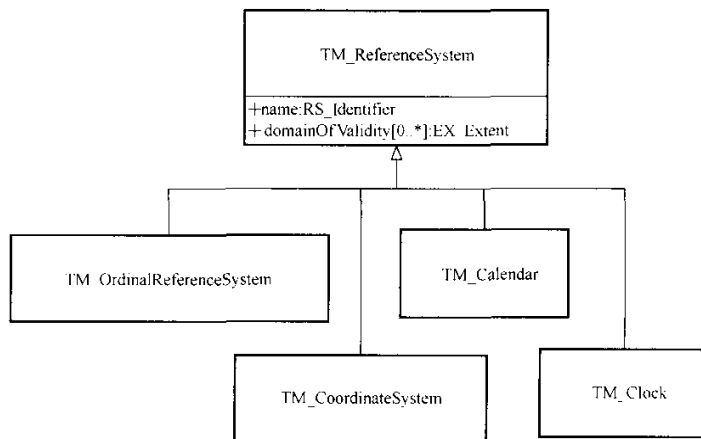


图 7 时间参照系

TM_ReferenceSystem 类应提供下列属性：

- a) name:RS_Identifier 应提供一个唯一标识该时间参照系的名称。数据类型 RS_Identifier 在 ISO 19111 中定义。
- b) DomainOfValidity:EX_Extent 应标识应用该 TM_ReferenceSystem 的空间和时间。数据类型 EX_Extent 在 ISO/TS 19103 中定义。它允许描述空间和时间两者的范围。当应用模式包括基于一个 TM_ReferenceSystem 的 TM_TemporalPosition,且 TM_ReferenceSystem 的有效范围小于数据集包括的时间和空间值的范围时应使用该属性。

5.3.2~5.3.4 描述三种参照系类型的模式。

5.3.2 日历与时钟

5.3.2.1 概述

日历和时钟两者均基于间隔标度。日历是一个不连续的时间参照系,它提供了分辨率为一日的定义时间位置的基础。时钟提供定义 1 日内的时间位置基础。为对特定日内的时间位置提供完整的描述,时钟必须与日历一同使用。图 8 提供 TM_Calendar(TM_日历)与 TM_Clock(TM_时钟)类的详细描述。

日历有各种复杂的内部结构。本模式定义一个简单的外部日历接口。附录 D 提供日历内部结构的详细描述。

每种日历提供依据一系列元素,如:年、月、日构成日历日期的一系列规则。每种日历中,年的记数相对于定义一个日历时代参考事件的日期。一个日历可能参考一个以上日历时代(见 D.3.1 和 D.3.2 的示例)。

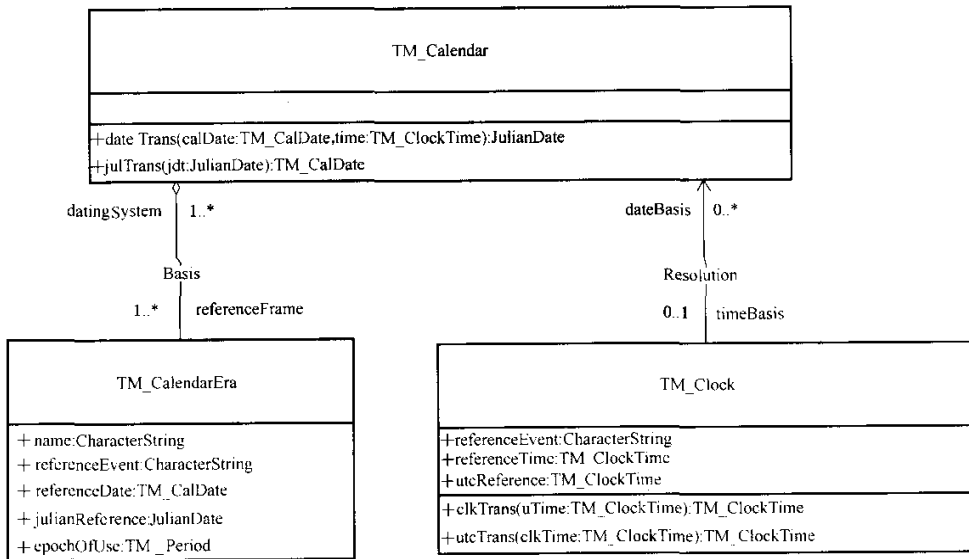


图 8 日历与时钟

5.3.2.2 日历时代

TM_CalendarEra(TM_日历时代)类应包括下列属性：

- a) name:CharacterString 应唯一标识该日历中的日历时代。
- b) referenceEvent:CharacterString 应提供历史的或神话的事件的名称或描述,该事件固定了该日历时代的基础标度的位置。
- c) referenceDate:TM_CalDate(TM_日历日期)应提供该参考事件的日期,该日期在给定的日历中被表示为日期。在大多数日历中,该日期是标度的原点(例如:第一日),但并不总是如此。

- d) julianReference: JulianDate 应提供相对于参考日期的儒略日期。
- e) epochOfUse: TM_Period 应标识日历时代被用作的日记数基础的 TM_Period。用于 TM_Period.begin 和 TM_Period.end 的数据类型应是 JulianDate (5.4.5.2)。

关联:

- a) Basis 应将这个 TM_CalendarEra 与用这个 TM_CalendarEra 作为日记数参照的那些 TM_Calendar 相连。

5.3.2.3 日历

TM_Calendar 应支持下列操作:

- a) dateTrans(calDate: TM_CalDate, time: TM_ClockTime): JulianDate 应接受一个特定日历的日期和一个特定时钟的时间作为输入,并返回一个儒略日期。
- b) julTrans(jdt: JulianDate): TM_CalDate 应接受一个儒略日期作为输入,并返回一个该日历日期。

注:儒略日记数系统是一个时间坐标系统,它具有早于任何已知日历的原点。日历日期与儒略日期的相互转换,为从一个日历日期转换为另一个日历日期提供了一个相对简单的基础。

一个特定日历的内部结构的任何描述应包括使用户可以实现这些操作的充分信息。它应包括与该日历相关的每一个日历时代描述,还应提供使该日历日期映射到等效的儒略日期的足够信息。

关联:

- a) Basis 应将这个 TM_Calendar 连接到作为日记数参考的 TM_CalendarEra。
- b) Resolution 应将这个 TM_Calendar 连接到用于指定在最小日历间隔内的时间位置的 TM_Clock。

5.3.2.4 时钟

TM_Clock 应包括下列属性:

- a) referenceEvent: CharacterString 应提供确定时钟基本标度位置的事件的名称或描述,例如:正午或日出。
- b) referenceTime: TM_ClockTime 应提供与该参考事件相关的日时间,表示为给定时钟的日时间。参考时间通常是时钟标度的原点。
- c) utcReference: TM_ClockTime 应提供相对于参考时间的当地时间或 UTC。

TM_Clock 应支持下列操作:

- a) clkTrans(UTime: TM_ClockTime): TM_ClockTime 应接受当地时间或 UTC,返回根据特定时钟表示的等效日时间。
- b) utcTrans(clkTime: TM_ClockTime): TM_ClockTime 应接受根据特定时钟表示的日时间,返回当地时间或 UTC 的等效时间。

5.3.3 时间坐标系

以日历日期和日内时间规定时间位置使点之间的距离计算和时间操作的功能描述复杂化。时间坐标系可以支持这类应用。时间坐标系建立在用单一时间间隔定义的连续间隔标度的基础上。

TM_CoordinateSystem(TM_坐标系)(见图 9)包括两个属性:

- a) origin: DateTime 应提供标度原点。原点应用带有 UTC 日时间的公历规定。DateTime 在适当的分辨率级别上可以截短。
- b) interval: CharacterString 应返回一个作为标度基础的时间间隔度量单位名称。根据应用可以选择适当的时间间隔,但是它应是由 GB 3102.1-1993 规定的各种时间度量单位之一,或是由 GB 3100-1993 描述的是上述单位之一的倍数。

TM_CoordinateSystem 应支持下列两个操作:

- a) transformCoord(c_value: TM_Coordinate): DateTime 应接受这个时间坐标系内的一个坐标值,并返回在公历和 UTC 中的等效 DateTime。

- b) transformDateTime (dateTime:DateTime): TM_Coordinate 接受公历和 UTC 的日期时间, 并返回一个等效的 TM_Coordinate。

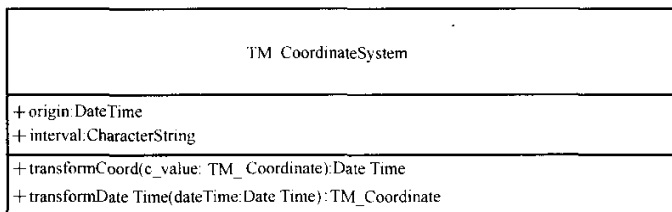


图 9 时间坐标系

5.3.4 顺序时间参照系

在大量的地理信息的应用中,例如:地质和考古,认为时间的相对位置比持续时间更准确。事件时间的相对位置可以充分确定,但是事件之间时间间隔的量不易准确确定。这种情况应采用顺序时间参照系。

顺序时间参照系基于顺序标度。在最简单的情况下,顺序时间参照系是事件的一个顺序序列。通常,事件的一个特定序列与一个位置相关联。不同位置之间的时间关系只能描述到一定程度,即基于事件的非时间特性将一个位置的事件与另一个位置的事件对比。在类似的已经发生的事件中就时段的定义而言,这种对比可以用于形成更广基础的时间参照系。在本标准使用的术语“顺序时代”就属于时段。

顺序时间参照系由一系列顺序时代构成(图 10)。顺序参照系通常具有层次结构,在给定的层级中顺序时代包括一个相连的较短的顺序时代的序列。

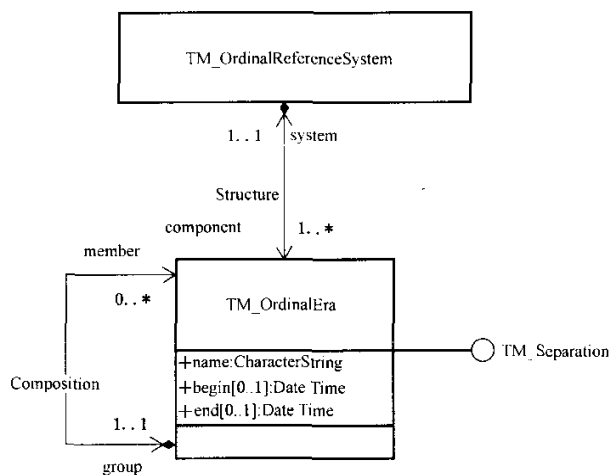


图 10 顺序时间参照系

TM_OrdinalReferenceSystem 仅提供从 TM_ReferenceSystem 继承来的属性。关联的结构指向 TM_OrdinalEras(TM_顺序时代)序列,它构成继承的最高级。

TM_OrdinalEras 包括 3 个属性:

- name:CharacterString 在 TM_OrdinalReferenceSystem 中唯一标识顺序时代。
- begin:DateTime 可提供在顺序时代开始时的时间位置,如果该时间位置是已知的。TM_TemporalPosition 在以公历和世界协调时日时间中应被指定为 DateTime。该 DateTime 可被截短为由 GB/T 7408—2005 规定的适当的分辨率级。
- end:DateTime 可提供在顺序时代结束时的时间位置,如果该时间位置是已知的。TM_TemporalPosition 在以公历和世界协调时日时间中应被指定为 Datetime。该 DateTime 可以被截短为由 GB/T 7408—2005 规定的适当的分辨率级。

TM_OrdinalEras 可支持 TM_Separation 接口(见 5.2.3.5)。

TM_OrdinalEras 可由关联组成标识的更短的 TM_OrdinalEras 序列构成。

5.4 时间位置

5.4.1 概述

标识时间位置的方法对每个时间参照系的类型都是特定的。适合于地理信息应用的参照系由公历和 UTC 结合而成(5.3)。ISO/TS 19103 定义的数据类型可用于将日期表示为符合 GB/T 7408—2005 的字符串。GB/T 7408—2005 规定了公历和 UTC 的使用。本标准定义用于在其他时间参照系中规定时间位置的数据类型。

5.4.2 TM_Position

TM_Position 是一个联合类,由列为其属性的数据类型之一构成。Date、Time 和 DateTime 是在 ISO/TS 19103 中定义的基本数据类型。它们遵照 GB/T 7408—2005 将日期和时间编码为字符串。这些数据类型可以用于描述基于公历和 UTC 的时间位置。TM_TemporalPosition 及其子类应用于描述基于其他时间参照系的时间位置。5.4.4 定义的数据类型规定日期和时间的数值,它们可用于参照任何日历和时钟的时间位置,包括公历和 UTC。

5.4.3 TM_TemporalPosition

TM_TemporalPosition 有 4 个子类(图 11),一个同 TM_ReferenceSystem 的关联和一个属性。属性:

- a) indeterminatePosition: TM_IndeterminateValue(TM_不确定值)是一个可选属性。除非 TM_TemporalPosition 的子类型被用做数据类型,该属性只提供 TM_TemporalPosition 值。当该属性同一个 TM_Position 子类型一起使用时,它为子类型所提供的的时间位置特定值提供一个限定词。

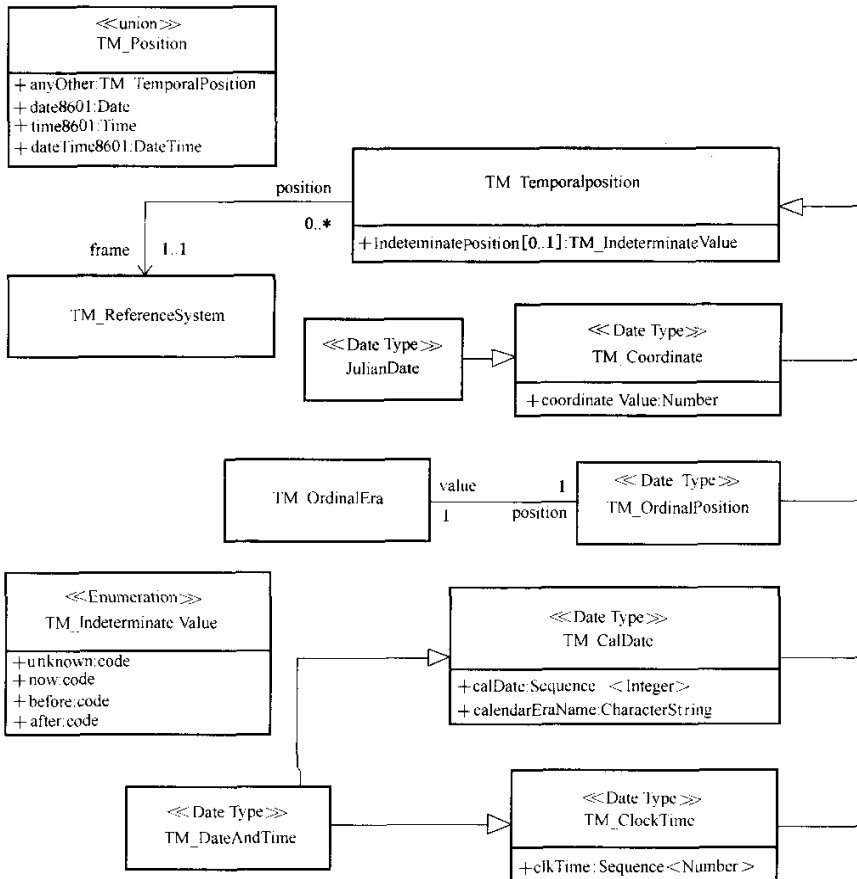


图 11 时间位置的数据类型

枚举数据类型 `TM_IndeterminateValue` 提供 4 个不确定位置的值：

- a) “unknown”应同父类 `TM_TemporalPosition` 一起使用，表明没有为时间位置提供特定的值。
- b) “now”应同 `TM_TemporalPosition` 的任一子类型一起使用，表明每当该值被存取时特定的值应被当前时间位置所更替。
- c) “before”应同 `TM_TemporalPosition` 的任一子类型一起使用，表明实际的时间位置是未知的，但是知道是在特定时间之前。
- d) “after”应同 `TM_TemporalPosition` 的任一子类型一起使用，表明实际的时间位置是未知的，但是知道是在特定时间之后。

关联：

- a) 关联 `Reference` 将 `TM_TemporalPosition` 连接到 `TM_ReferenceSystem`。每一个 `TM_TemporalPosition` 都应同与一个 `TM_ReferenceSystem` 关联。该关联在实例级不必明确。如果不说明，则是指同公历和 UTC 的关联（见 5.3.1）。它也可以在要素目录的属性类型定义中或在数据集的元数据中标识。

5.4.4 基于日历和时钟的位置

5.4.4.1 日历日期

`TM_CalDate` 是数据类型，用于标识日历中的时间位置。`TM_CalDate` 有 2 个属性：

- a) `CalendarEraName:CharacterString` 提供该日期参照的日历时代的名称。
- b) `calDate:Sequence<Integer>` 提供一个正整数序列，在该序列中第一个整数标识日历最高层次使用的单位的特定实例，第二个整数指明日历次高级层次上使用的单位的特定实例，依次类推。GB/T 7408—2005 中定义的公历日期格式可以用于以年、月、日构成的任何日期。

示例：在公历中序列 1999,09,03 表明 1999 年第 9 个月第 3 日的时间位置。按 GB/T 7408—2005 格式可表示为 19990903。

5.4.4.2 时钟时间

`TM_ClockTime` 是用于标识一日内时间位置的数据类型。由于 `TM_TemporalPosition` 本身不能完整地标识单一时间位置，因此它应同 `TM_CalDate` 一起使用。它也可以标识每日重复发生的事件出现的时间。`TM_ClockTime` 有 1 个属性：

- a) `clkTime:Sequence<Number>` 提供一个类似于 `calDate` 结构的正数序列。第一个整数表明时钟最高层次上使用的单位的特定实例，第二个数表明时钟次高层次上使用的单位的特定实例，依次类推。在序列中除最后一个数外所有数都应是整数，最后一个数可以是整数或者是实数。

示例：在近代 24 小时时间序列中，22, 15, 30.5 定义的是第 22 点第 15 分开始后 30.5 秒的时间位置。在 GB/T 7408—2005 格式中可表示为 221530.5。

5.4.4.3 日历日期和时钟时间

`TM_DateAndTime(TM_日期和时间)` 是 `TM_CalDate` 和 `TM_ClockTime` 两者的子类型。为提供用于识别小于一日分辨率的时间位置的单一数据类型，它继承两者的属性。

5.4.5 基于时间坐标系的位置

5.4.5.1 TM_Coordinate

`TM_Coordinate` 是一个在时间坐标系中表示时间位置的数据类型。`TM_Coordinate` 有一个属性：

- a) `CoordinateValue:Number` 保存距标度原点的距离，表示为与该时间坐标系相关联的标准时间间隔的倍数。

5.4.5.2 儒略日期

儒略日记数系统是一个时间坐标系，约定儒略历起算点为公元前 4713 年 1 月 1 日格林尼治平正午。儒略日数是一个整数，儒略日期是一个允许更高分辨率的小数值。

5.4.6 基于顺序时间参照系的位置

在顺序时间参照系中,时刻的时间位置是它发生在最短的顺序时代(在最低层)的位置。TM_OrdinalPosition(TM_时间顺序位置)是应用于表明在顺序时间参照系中时间位置的数据类型。TM_OrdinalPosition 有 1 个属性:

- a) ordinalPosition:Reference<TM_OrdinalEras> 提供时刻发生的顺序时代的参照。

示例:表 1 是地质年代表的一部分,它是一个顺序时间参照系。在新生代中某个时刻的时间位置应用该时刻所在的世标识;而在中生代某一时刻的时间位置应用它所在的纪标识。

表 1 地质年代表部分

代	纪	世
新生代	第四纪	全新世
		更新世
	第三纪	上新世
		中新世
		渐新世
		始新世
		古新世
中生代	白垩纪	
	侏罗纪	
	三叠纪	

5.5 时间和地理信息组分

5.5.1 地理信息组分的时间内容

ISO 19109 描述的通用要素模型提供了地理信息元模型。它标识显示在图 12 中的要素属性、要素操作和要素关联的元类。每一组分都可包括时间内容。ISO 19109 提供了在应用模式中将这元类的实例化的规则。5.5.2~ 5.5.4 描述在通用要素模型中某些元类的时间子类。当这些组分涉及到时间信息时,它们也提供使用的扩充规则。附录 B 提供了在应用模式中使用这些类和规则的方法示例。

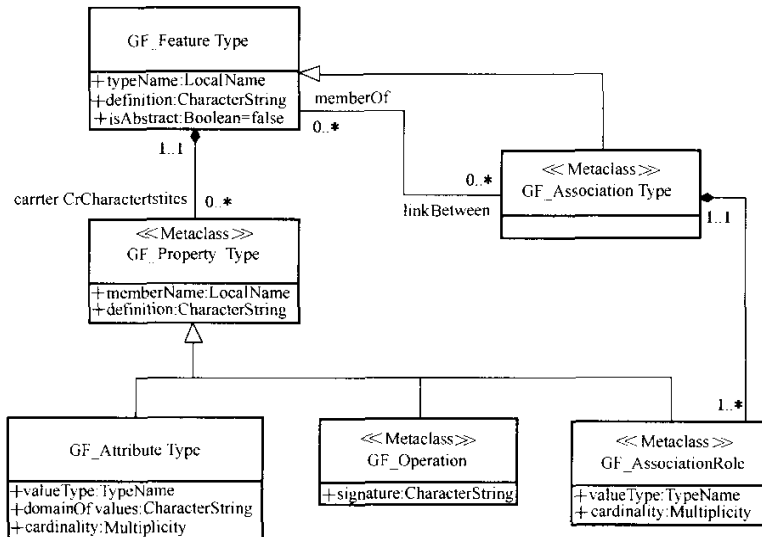


图 12 通用要素模型的元类

ISO 19110 规定用于要素编目的规则,包括要素类型、要素属性、要素操作和要素关联的定义。

5.5.2~5.5.4 规定了定义组分中含有时间内容时的要求。

GB/T 19710—2005 规定用于地理信息的元数据元素的标准集,并提供一个用于定义适合于特定的应用或专用标准的增加元数据元素的机制。本标准的 5.5.5 提出用于定义包括时间内容的元数据元素的要求。

5.5.2 时间要素属性

时间要素属性描述与时间位置有关的要素特性。与 ISO 19109 中定义的应用模式相协调,GF_TemporalAttributeType(GF_时间属性类型)通常应在应用模式中实例化为一个类的属性,该属性本身是 GF_FeatureType(GF_要素类型)元类的一个实例。在 ISO 19109 说明的条件下,它可以实例化为一个 UML 类,该类与其作为属性的要素类型类相关联。

静态时间特性有两类:事件和状态(图 13)。

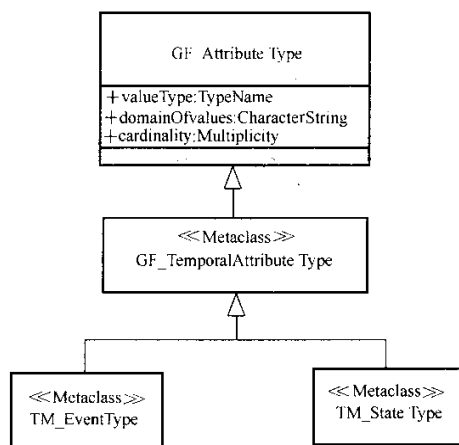


图 13 时间要素属性类型

- 事件是发生在某一时刻的行为。事实上,几乎每一个事件都占有一个短暂的时间间隔,但是,当间隔小于度量标度的分辨率时,可表示为时刻。一个事件的 GF_AttributeType.valueType 或是 TM_Instant、TM_Node 或是 TM_TemporalPosition。
- 状态是一个条件——持续一个时段的要素或数据集的一种特性。该特性可以由要素属性或元数据元素表示。描述状态的 GF_TemporalAttributeType 可以用两种方式实例化。在简单情况下,GF_AttributeType 应实例化为表示要素类型的类的属性,其 GF_AttributeType.valueType 应是 TM_Period 或 TM_Edge。当需要更多信息时,表示状态的 GF_TemporalAttributeType 应被实例化为另一个 UML 类。该类应是 TM_Period 的子类型,它继承 Beginning 和 Ending 的关联,或者它是 TM_Edge 的子类型,继承 Initiation 和 Termination 的关联。该状态的特性应由该类的一个或多个属性描述。该类的重现应由与之关联的要素类型类的属性端的多重性表示。通常,状态的变化与启动或终止该状态的事件有关。该事件应由表示该状态的类的属性标识。

事件可以在多个时刻重现,状态也可以在多个时间重现。FeatureAttributeType. Cardinality 应规定应用模式允许重现的次数。

事件或状态通常基于规则的时间基础重现。两个连续发生的事件或状态之间的间隔持续时间是它的周期时间。当时间要素属性描述重复发生的现象时,在应用模式中应被描述为一个 UML 类,该类与要素类型类相关联。该类至少应有两个属性:一个标识事件发生的特定时刻或属性专题值存在的持续时段,另一个标识事件或状态出现之间的时段时间。数据类型 TM_Duration (见 5.2.3.7)用于表示时段时间的值。

ISO 19110 规定应包括在要素目录的要素属性描述中的元素。当 GF_TemporalAttributeType 是一个事件时,该属性的要素属性定义 (Feature Attribute Definition) 应标识其行为和说明其时间位置的分辨率。当 GF_TemporalAttributeType 是一个状态时,该属性的要素属性定义将标识该状态的特性。对于时间要素属性,要素属性值域类型是“不能枚举的”的。要素属性值的值域、要素属性值数据类型和要素属性值度量单位由时间参照系定义,该参照系是描述时刻和时段的时间位置的基础。如果时间参照系不是由公历和 UTC 的结合形成,要素属性值值域的描述应定义使用的时间参照系。要素属性值的数据类型和要素属性值的度量单位都应与时间参照系一致。

5.5.3 时间要素操作

时间要素操作是动态的,它们描述要素一个或多个方面的值随时间函数变化的方式。时间要素操作在应用模式中应实例化为要素类型类的操作。对于该要素类型类,时间是作为 GF_Operation. signature 的一个输入参数。

示例:给出一个要素类型类 RoadIntersection,通过该交叉口的交通流可由 UML 操作描述为:

TrafficFlow (tMin:Integer, tMax:Integer,time:TM_ClockTime):Integer

ISO 19110 规定定义要素操作的要求。对于时间要素操作,要素操作文本描述 (Feature Operation Textual Description) 应描述操作变化的返回值随时间函数变化的方式以及它是如何受时间特性影响的。要素操作的形式化定义 (Feature Operation Formal Definition) 应包括作为独立变量之一的的时间。

5.5.4 时间和要素的关联

5.5.4.1 要素关联的时间方面

要素关联有两种方式涉及时间。一些要素关联存在是由于有关要素实例的时间特性。另一些可能由于多种原因而存在的要素关联,拥有它们自己的作为关联的时间特性。

示例:同一时间存在的两个要素之间的关联说明了第一种情况。该时间关联没有其他特性,并不依赖所涉及的要 素类型。第二种情况的示例是在同一时段内属同一所有者的两片土地之间的关联。在这种情况下,关系的主导元素是共同所有者,时间元素是该关联的次要特性。

时间要素关联是第一种类型的相关……即为要素实例的时间特性而存在的关联。

5.5.4.2 时间要素关联

5.5.4.2.1 概述

时间要素关联是由该关联连接的要素生存期之间的关联的显式描述。如果每个要素都有描述其生存期的属性,则有可能用 TM_Order.relativePosition 操作比较两个要素的生存期导出时间要素关联。如果在应用中时间要素关联是重要的,该应用模式应为每一个它规定的要素类型指定一 lifeSpan 属性。当时间位置基于日历与时钟或时间坐标参照系时可使用属性 lifeSpan:TM_Period。当时间位置基于顺序时间参照系时,应使用属性 lifeSpan:TM_Edge,应用模式应将 TM_TopologicalComplex 实例化为非线性图,该图包括表示生存期的所有 TM_Edge。对于不支持 TM_Order 接口的应用应规定显式时间

要素关联。

本标准规定时间要素关系的两个子类型:简单时间要素关联和继承(见图 14)。

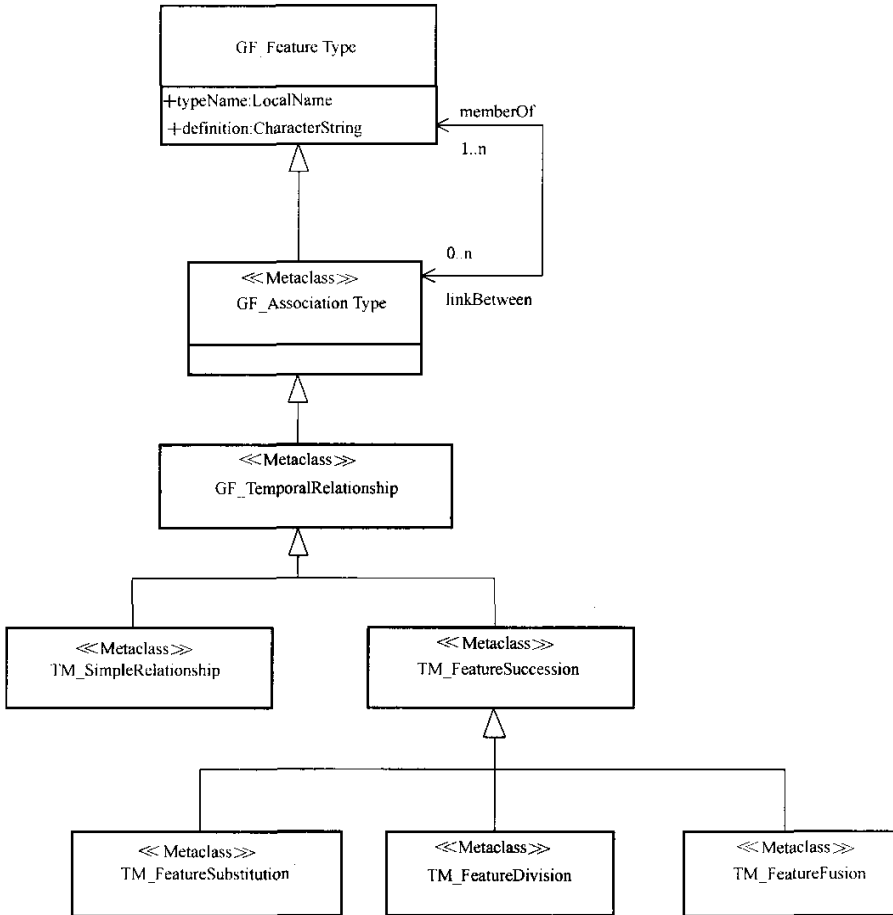


图 14 时间要素关联分类

5.5.4.2.2 简单时间要素关联

简单时间要素关联标识两个或多个要素的时间相对位置。原则上,在所有要素实例之间存在这种关联类型,但是,不能断定所有特定要素之间都存在这种关联类型。

根据 ISO 19109 定义的应用模式规则,要素关联通常应实例化为表现要素类型的 UML 类之间的关联。但是,纯粹的时间要素关联通常独立于要素类型。如果应用只涉及某些要素类型之间的时间要素关联,则可将其实例化为要素类型类之间的关联。对于要求保持所有要素类型实例之间的某种时间关联类型的应用模式是低效率的,比较好的情况是应用模式支持 5.5.4.2.1 描述的那些关联的派生。一个可供选择的办法是将本模式中所有要素类型类的超类的要素类实例化,并将该层次上的要素关联实例化为自相关的关联。

5.5.4.2.3 要素更替

要素更替是一个要素实例集由另一个要素实例集取代。取代意味着第一个要素实例集的生存期在

第二个要素实例集生存期开始的这个时刻结束。要素更替有空间和时间 2 个方面的内容,在该关联中,要素实例在不同的时间以特定的顺序占据相同空间位置。

要素更替不总是类型依赖的,这就是说,被更替的要素实例的类型并不总能从那个更替它的要素实例类型预知。要素更替可以在基本要素层次上建模,而不总是在要素类型层次上建模。

有三类要素更替:要素置换,要素分割和要素融合。要素置换是一个要素实例由另一个相同或不同要素类型的要素实例取代。它在两个要素实例之间建立一个一对一的关联。要素分割指将单一要素分离为两个或多个相同类型的要素实例。它在要素实例之间建立一对多的关联。要素融合发生在两个或多个相同类型的要素实例融合为单一要素实例时。它在要素实例之间建立多对一的关联。一个单一事件可以导致上述类型组合的更替形式。可能的四种组合是:

- 分割和置换;
- 融合和分割;
- 融合、置换和分割;
- 置换和融合。

示例:林地中的空旷部分由停车场替代是“分割和置换”的示例。林地的空旷地与邻接的牧场融合为牧场是“置换和融合”的示例。

注:单一要素本身特性的变化不属要素更替。例如,考虑要素类型 Building 有一个属性 numberOfOccupants。该属性值可能定期更新,但不能认为是一个建筑物实例由另一个建筑物实例取代。一个要素类型的实例被认为取代了一个较早的相同要素类型实例所必须的变化程度依赖于应用。作为一般规则,当要素标识符改变时可认为有取代发生。

更替关联可通过使用 TM_Order.relativePosition 和由 ISO 19107 定义的一个或多个空间操作导出。

在应用模式中,要素更替类型的时间要素关联可实例化为要素类型类之间的 UML 关联或作为一个基本要素类的自相关关联。每个更替类型的名称、角色和多重性是不同的。角色的名称应指明一个要素更替另一个的顺序。为包括更替发生的时间,应用模式应将更替关联表示为一个带有标识其出现时间属性的 UML 关联类。

5.5.4.3 要素关联的时间特性

要素关联的时间特性通常提供关联开始和结束的时刻或其持续时段的信息。像时间要素属性或时间元数据元素一样,它们描述事件或状态。具有时间特性的要素关联应实例化为 UML 关联类。时间特性应表示为实例化 TM_Event(TM_事件)或 TM_State(TM_状态)元类(5.5.2)的属性。

5.5.5 时间元数据元素

GB/T 19710—2005 定义了地理信息的标准元数据元素集。它也规定了在应用模式(图 15)中定义附加元数据元素的方法。时间元数据元素与时间要素属性类似,两者都描述同时间位置关联的事件或状态的静态特性。其不同之处在于要素属性描述一个现实世界对象要素实例中被抽象的特性,而元数据元素描述数据的特性。一个元数据元素描述的数据范围可从数据集集合到要素的单一特性。

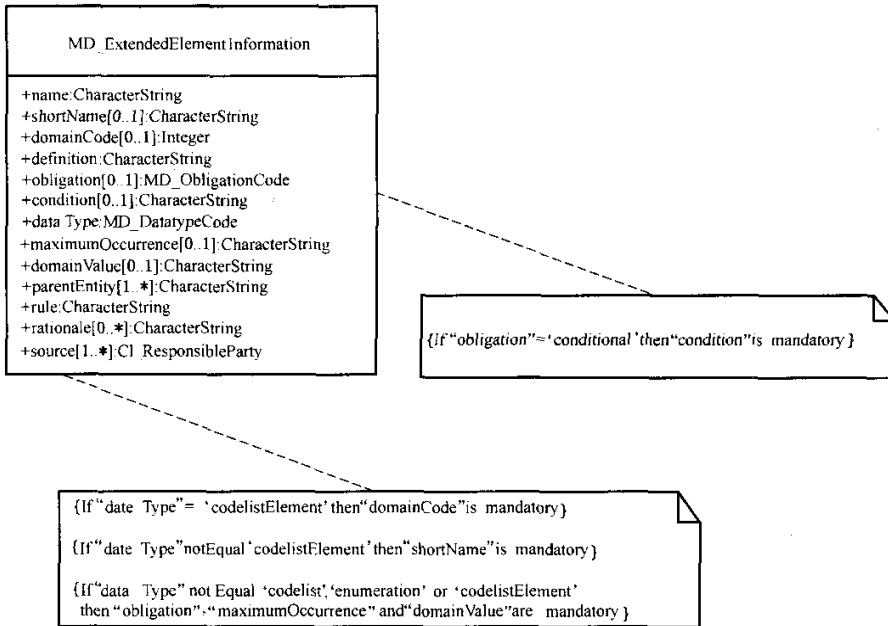


图 15 扩展的元数据元素

当一个时间元数据元素描述一个事件时，其名称和定义应标识行为和被规定的时间位置的分辨率。元数据元素的值是 `TM_TemporalPosition` 的 `TM_Instant` 的实例，或 `TM_TemporalPosition` 的子类型之一。

当时间元数据元素描述状态时，名称和定义应描述状态的特性。元数据元素的值应是 `TM_Period`。

如果元数据元素不是基于公历和 UTC 的组合，定义应标识所使用的时间参照系。

附 录 A
(规范性附录)
抽象测试套件

A.1 数据传输应用模式

- a) 测试目的:检验数据传输应用模式定义要素的时间属性、要素的时间关联和时间元数据元素是否遵守规定的要求。
- b) 测试方法:检查包括在应用模式中的要素时间属性和时间关联的表达,保证其定义满足表示时间属性或其值的时间对象的使用要求;保证在应用模式中定义的任一时间元数据元素满足要求;保证时间对象所有必需属性和关联以及可选属性和关联按要求实现。检验用于时间位置数据类型的值。
- c) 引用:5.2、5.4、5.5.2、5.5.4.2 和 5.5.5。
- d) 测试类型:基本测试。

A.2 带操作的数据应用模式

- a) 测试目的:检验支持对数据操作的应用模式是否满足 A.1 的要求,并按规定和要求,定义时间要素操作,实现时间对象操作。
- b) 测试方法:检查包括在应用模式中的时间要素操作的表示,保证它们满足要求。检查包括在应用模式中的要素时间属性的表示,保证用于表示时间属性及其值的时间对象的任何操作按要求实现。
- c) 引用:A.1、5.2、5.5.3 和 5.5。
- d) 测试类型:基本测试。

A.3 要素目录

- a) 测试目的:检验要素目录是否按规定的要求定义时间特性。
- b) 测试方法:检查目录。核实时间要素属性、时间要素操作或时间要素关联的定义符合要求。保证满足标识和描述时间参照系的要求。
- c) 引用:5.3、5.5.2、5.5.3 和 5.5.4.2。
- d) 测试类型:基本测试。

A.4 元数据元素规范

- a) 测试目的:检验时间元数据元素的定义是否满足规定的要求。
- b) 测试方法:检查元数据规范中的元数据元素定义,保证定义满足事件和状态的要求。
- c) 引用:5.5.5。
- d) 测试类型:基本测试。

A.5 数据集元数据

- a) 测试目的:检验数据集的元数据是否提供有关时间参照系的必要信息。
- b) 测试方法:检查元数据集。如果元数据没有表明包括在它描述的数据中的所有时间位置都基

于公历和 UTC, 检查看它是否提供了数据采用的每一个时间参照系的说明或其出处, 核实每项描述保证它包括所有需要的元素。

- c) 引用: 5.3。
- d) 测试类型: 基本测试。

附录 B

(资料性附录)

应用模式中时间的使用

B.1 时间要素属性

B.1.1 TM_GeometricPrimitive 作为数据类型

图 B.1 说明在应用模式中 TM_GeometricPrimitive 的使用。要素类型 Building 被表示为 UML 类。它有一个 dateOfConstruction 的属性, 该属性有一个数据类型为 TM_Instant 的值; 其第二个属性称为 periodOfOccupancy, 有一个数据类型为 TM_Period 的值。在本模式中, TM_GeometricPrimitive 不执行在 TM_Order 和 TM_Separation 接口定义的操作。由于 TM_Instant 仅有一个属性 TM_Instant.position; TM_TemporalPosition, 它取数据类型 TM_TemporalPosition 的值, TM_Position 数据类型或它的子类, 例如 DateTime, 都可以作为 Building.dateOfConstruction 的数据类型。

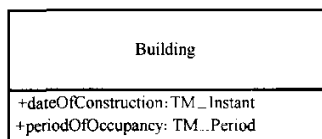


图 B.1 TM_GeometricPrimitive 作为数据类型

B.1.2 TM_GeometricPrimitive 作为时间属性

图 B.2 说明用于时间要素属性的 TM_GeometricPrimitive 的一个可供选择的使用方式。在这种情况下, 要素属性 periodOfOccupancy 表示为一个 UML 类, 该类由一个 UML 关联连接到 Building。PeriodOfOccupancy 是 TM_Period 的子类型。它从 TM_Period 继承关联角色开始和结束, 但是它将每种情况的数据类型都限制为 DateTime。[在 TM_Period 级, 这些属性的数据类型是 TM_Position]。它还继承了 TM_Order 接口和 TM_Separation 接口。通过前者使用操作 relativePosition(other: TM_GeometricPrimitive); TM_RelativePosition, 通过后者使用操作 Length(); TM_Duration 和 distance(other: TM_GeometricPrimitive); TM_Duration。

B.1.3 TM_TopologicalComplex 作为属性

图 B.3 说明 TM_TopologicalComplex 作为时间要素属性时的使用。一个考古遗址的历史通常用某文化团体的一些成员占据或不占据该遗址的一系列时段描述。这些时段的顺序是已知的, 但是时间位置是未知的。该历史可描述为 TM_TopologicalComplex。ArchaeologicalSite 表示为 UML 类, 同表

示要素属性 SiteHistory 的另一个 UML 类相关联。SiteHistory 是 TM_TopologicalComplex 的子类型。它是 OccupancyIntervals 的聚集,其中每个 OccupancyIntervals 是 TM_Edge 的子类型。

B.1.4 属性值的循环重现

图 B.4 提供要素属性具有循环重现值的示例。要素类型 Field 有一个属性 CropRotation,它被表示为多重性为 0 对 n 的 UML 类。在该田地种植的每类作物的该属性类型都有一个实例。农作物复种有三个属性:

- cropType 表示作物的一种类型;
- firstPlanting 表示在该田地该作物首次种植时间;
- rotationInterval 表示该作物不同种植之间的周期。

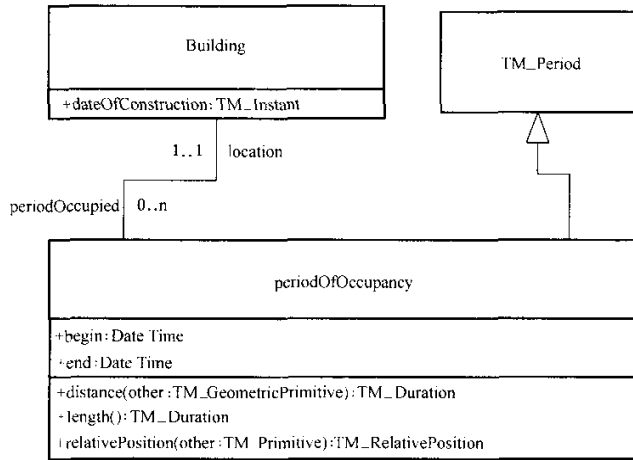


图 B.2 TM_GeometricPrimitive 作为时间要素属性

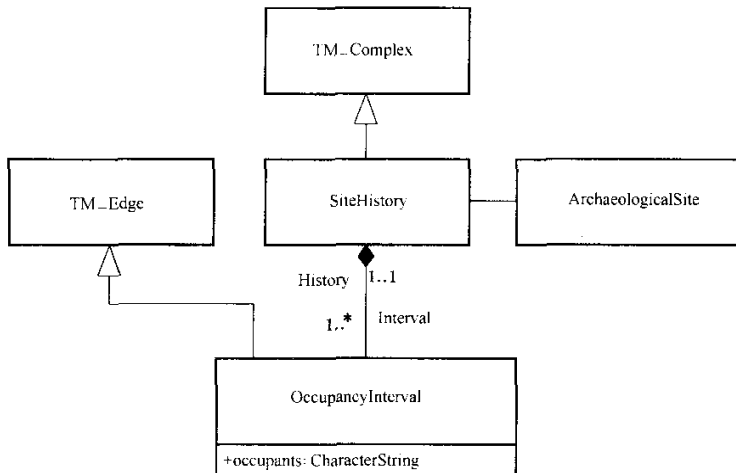


图 B.3 TM_TopologicalComplex 作为时间要素属性

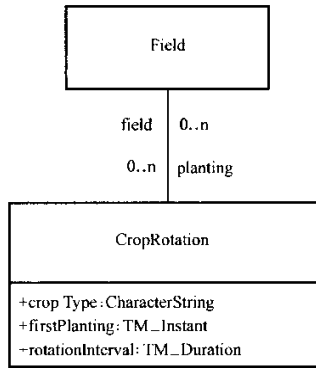


图 B.4 循环属性值

B.2 时间要素关联

B.2.1 简单时间关联

图 B.5 是作为 UML 关联实现的简单要素关联的示例。该模式指明道路在服务站存在之前存在。

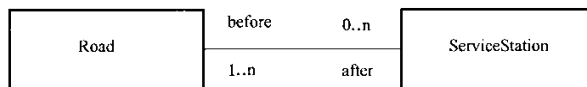


图 B.5 简单时间关联显式表示

图 B.6 表示建筑物的简单模式，它提供要求通过使用操作 TM_Order.relativePosition 派生简单要素关联的数据实例。

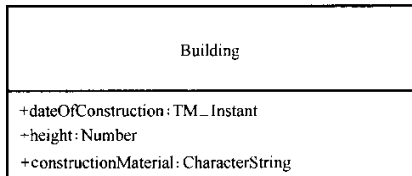


图 B.6 建筑物模式

假定根据该模式建立的一个数据集包括 4 个具有表 B.1 所示属性值的 Building 实例。

表 B.1 建筑物模式数据集

Building	constructionMaterial	height	dateOfConstruction:TM_Instant, position [date]
A	砖	45 m	1982
B	钢铁	4 m	1967
C	木	8 m	1941
D	砖	30 m	1967

TM_Instant 可通过 TM_Order 接口使用操作 relativePosition(other: TM_Primitive): TM_RelativePosition。如果以每一个 Building. dateofConstruction: TM_Instant 作源、每个其他实例作目标执行该操作,将返回表 B. 2 所列的值。

表 B. 2 实例 Building 间的简单时间关联

源实例	目标实例			
	A	B	C	D
A	等于	之后	之后	之后
B	之前	等于	之后	等于
C	之前	之前	等于	之前
D	之前	等于	之后	等于

B. 2. 2 要素更替

图 B. 7 是在要素类型类之间建模为关系的要素更替的示例。这是一个在美国东部普遍的、被称为古老田地更替的生态更替类型的示例。这些类型发生在所示顺序的一个点,条件是该点不受干扰。这说明与在要素类型级的要素更替建模有关的问题。在任何时候,火灾、暴风雨或人类的干预可中断这种更替或将它移至早期阶段或以不可预知的要素类型取代存在的要素类型。

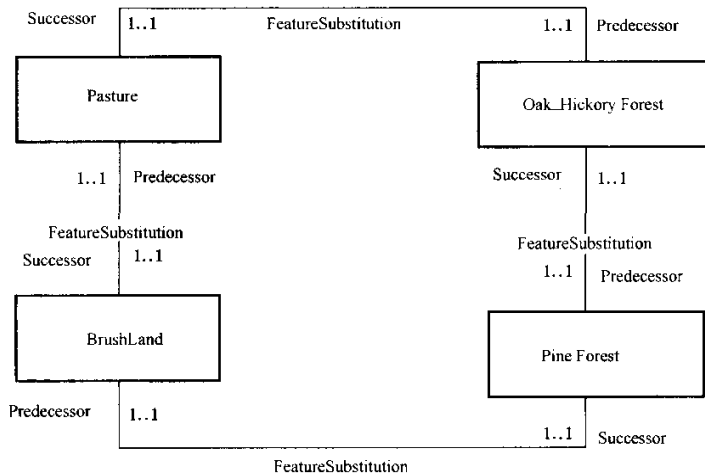


图 B. 7 要素类型之间的要素更替

图 B. 8 是建模为 UML 类的自相关关联的要素更替的示例,该 UML 类是可能涉及更替关联的多种要素类型的超类。由于没有办法预知这些要素类型实例彼此更替的顺序,因此以这种方式建模是必要的。在此例中,每个关联被表示为关联类,因此,更替发生的时间可以描述为该关联的属性。

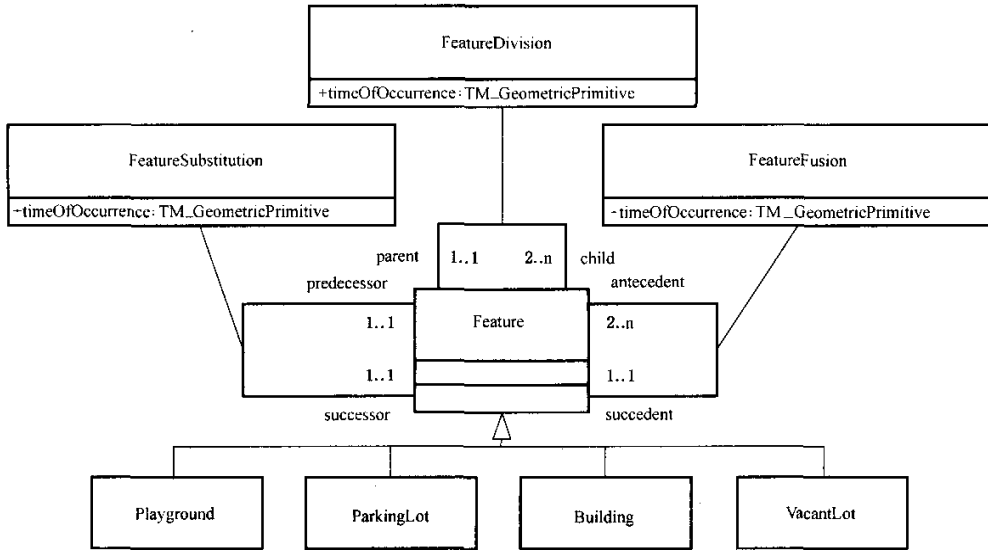


图 B.8 基本要素级的要素更替

B.3 具时间特性的要素关联

图 B.9 是牧场总部与其管理的出租牧场之间关联的应用模式。RangelandTract 通过租约 Lease 关联到 RanchHeadquarters。租约是一个不能认为是时间关联的要素关联的实例，因为时间并不是这个关联的主要方面。但是，租约只在时间的给定时段内有效。Lease 在该模型中被表示为允许给关联指定属性的 UML 关联类。此时，它有一个属性 leasePeriod 标识该租约的有效时段。它也可有其他属性，例如一个标识出租费用的属性。

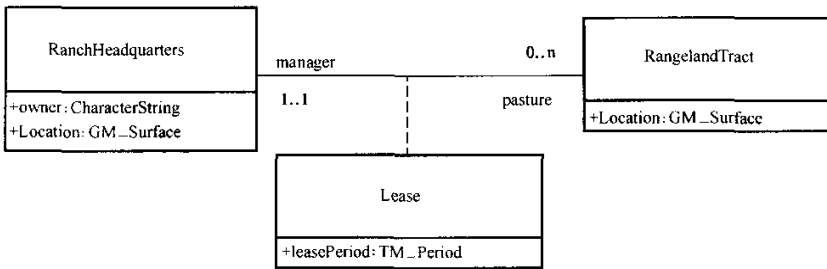


图 B.9 牧场租约

附 录 C

(规范性附录)

在元数据中描述时间参照系

C.1 时间参照系元数据

本标准 5.3.1 要求同使用非公历和 UTC 时间参照系的数据集相关联的元数据应提供描述那些时间参照系的文件的引用或者在元数据中包括那些时间参照系的描述。在表 C.1 中定义的元数据元素可用于此目的,表的结构遵循 GB/T 19710—2005 的附录 B。

表 C.1 用于描述时间参照系的元数据元素

	英文名称	中文名称	定义	约束/条件	最大出现次数	模型元素或数据类型	域
1	TM_Referenc-eSystem	TM_参照系	关于时间参照系的信息	C/数据集的时间信息不基于公历吗?	N	类	2~33 行
2	name	名称	已知时间参照系的名称	M	1	RS_标识符	ISO 19111
3	domainOfValidity	有效性范围	使用该时间参照系的空间和时间范围	C/数据集中使用的范围小于该时间参照系的范围吗?	N	EX_范围	ISO/TS 19103
4	Subtype	子类型	被描述的时间参照系子类型	M	1	特化 (Specialization)	“TM_Calendar” “TM_Clock” “TM_CoordinateSystem” “TM_Ordinal-ReferenceSystem”
5	TM_Calendar	TM_日历	日历的描述	C/是子类型“TM_Calendar”吗?	1	类	6~15 行
6	dateTrans	日期转换	转换指定日历日期到儒略日期的操作描述	M	1	字符串	自由文本
7	julTrans	儒略日期转换	转换儒略日期到指定日历日期的操作描述	M	1	字符串	自由文本
8	referenceFrame	参照框架	与被描述的日历关联的日历时代	M	N	角色名称	TM_CalendarEra

表 C.1(续)

	英文名称	中文名称	定义	约束/条件	最大出现次数	模型元素或数据类型	域
9	timeBasis	时间基础	与该日历一起用于定义一个日历日之内的时间位置的时钟	O	1	角色名称	TM_Clock
10	TM_Calendar-Era	TM_日历时代	每个日历时代的特性	M	N	类	11~15行
11	name	名称	该日历时代为已知名称	M	1	字符串	自由文本
12	referenceEvent	参考事件	用作这个日历时代基准的事件	M	1	字符串	自由文本
13	referenceDate	参考日期	所描述日历的参考事件日期	M	1	TM_CalDate	所描述日历中的日期
14	julianReference	儒略参考日期	参考事件的儒略日期	M	1	儒略日期	数字
15	epochOfUse	日历时代纪日基础的时段	该时代被用作日记数基础的时段	M	1	TM_Period	GB/T 7408
16	TM_Clock	TM_时钟	时钟的描述	C/Subtype “TM_Clock 或分辨率不为空吗?”	1	类	17~21行
17	referenceEvent	参考事件	用作该时钟基准的事件	M	1	字符串	自由文本
18	referenceTime	参考时间	用作该时钟参考事件的时间	M	1	TM_Clock-Time	所描述的时钟内的时间
19	utcReference	utc 参考	参考事件的 UTC 时间	M	1	TM_Clock-Time	GB/T 7408
20	utcTrans	utc 转换	该时钟时间转换到 UTC 时间的操作描述	M	1	字符串	自由文本
21	clkTrans	时钟转换	UTC 时间转换到该时钟时间的操作描述	M	1	字符串	自由文本
22	TM_CoordinateSystem	TM_坐标系	时间坐标系的描述	C/是子类型 “TM_CoordinateSystem”吗?	1	类	23~26行

表 C.1(续)

	英文名称	中文名称	定义	约束/条件	最大出现次数	模型元素或数据类型	域
23	origin	原点	该时标原点位置,它是时间坐标系的基础被表示为公历的日期和 UTC 日内时间	M	1	日期时间	GB/T 7408
24	interval	时间间隔	用于在该坐标系的坐标轴上度量持续时间的标准时间单位	M	1	字符串	GB 3102.1, GB 3100
25	transformCo- ord	转换坐标	将该时间坐标系坐标转换到公历日期和 UTC 时间的操作描述	M	1	字符串	自由文本
26	transformDate- Time	转换日期时间	将公历日期和 UTC 时间转换到该时间坐标系坐标的操作描述	M	1	字符串	自由文本
27	TM _ Ordinal- ReferenceSystem	TM _ 顺序参 照系	顺序时间参照系的描述	C/是子类型 “TM _ Coordi- nateSystem”吗?	1	类	28~33 行
28	component	组分	构成该顺序参照系的最高级别的顺序时代	M	1	角色名称	TM _ Ordinal- Era
29	TM _ Ordinal- Era	TM _ 顺序 时代	顺序时代的描述	M	N	类	30~33 行
30	name	名称	标识特定顺序时代的名称	M	1	字符串	自由文本
31	begin	开始	该顺序时代开始的日期	O	1	日期时间	GB/T 7408
32	end	结束	该顺序时代结束的日期	O	1	日期时间	GB/T 7408
33	member	成员	细分该顺序时代的顺序时代	M	1	角色名称	TM _ Ordinal- Era

注: 对应 UML 操作的元数据元素是该操作的文字描述。

附录 D
(资料性附录)
历法描述

D.1 历法的内部结构

日历是一个不连续的时间参照系,它提供定义一日分辨率时间位置的基础。公历是事实上的国际标准。它是地理信息应用的首选。但是,公历以外还有各种传统或历史的日历,可能适于某些地理信息的应用。例如,在所涉及的文化的日历中,考古物质可能更精确地测定日期。本标准 5.3.1 要求使用非公历的数据集元数据应包括所使用日历的描述或描述的引用。本附录说明在这样的描述中应考虑的内容。

日历具有层次结构(图 D.1),在该结构中每一层次使用一个特定的时间间隔。典型情况下,在结构中的一个层次的时间间隔实例以周期为基础命名或记数。该周期长度等于下一个较高层次使用的时间间隔。虽然大多数日历包括作为标准时间间隔的年和日,但有一些日历在中层次上没有使用月作为时间间隔。有些日历在层次结构中还包括其他层次,或甚至包括一个使用其他时间间隔的平行层次。日历标识该层次结构中每一层次的一个时间间隔的实例。

示例:公历中的一日标识为一个特定的年份,该年的特定月和该月的特定日。

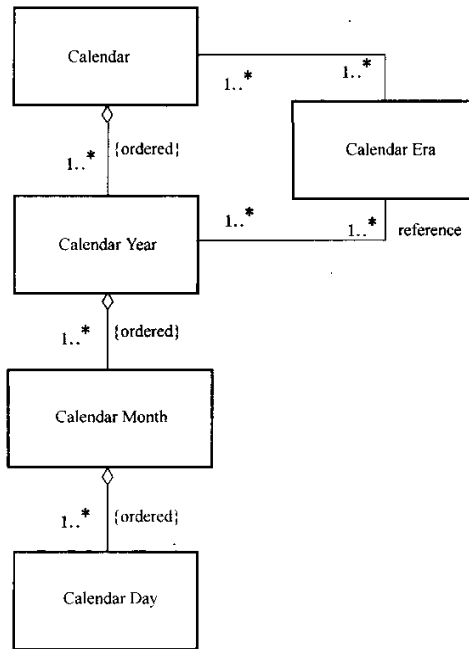


图 D.1 典型的日历内部结构

由于在名义上是基于天文周期,而天文周期的时段彼此之间不为整倍数,因此,日历是复杂的。为了维护日历周期和天文周期之间的相位关系,在低层次时间间隔置闰。

注:置闰是在日历中插入一个额外像日或月那样的时间间隔。例如,在公历中,每个闰年的2月底插入一个额外的日。

- a) 每一种日历的基本时间间隔是日,名义上日对应地球自转时段,在日历中使用的其他时间间隔长度是日的整倍数。在许多日历中,具体日由其在月中的顺序数确定。某些日历用其他方法。

- b) 名义上一个日历年相当于地球围绕太阳公转的周期。由于该时段的持续时间不是整日数,大多数日历设计为允许改变历年长度,使跨越长时段平均长度与地球公转周期相等。日历能够达到的精度不同。在大多数日历中,在基本循环中用额外的月或日置闰。在许多古老日历中,置闰是没有规则的,每当政府或宗教权威认为需要时就设置。
- c) 典型的年是由变长的顺序月构成的。有置闰规则时,年的内部结构遵循有限数量模式之一。每种模式用一个模版描述,该模版列出与其一致的年中月份的名称和长度。置闰规则提供一个时代中的特定年的序号与其所遵循模版相联系的机制。

月名义上对应月亮周期的时段或是月相周期或是月亮绕地球公转的周期。在许多古老日历中,月的长度由天文观测而不是由计算确定。月亮周期的时段不等于日长度的整数倍,天文年的时段也不等于任何月亮周期的整数倍。某些日历通过限定日历月长度变化的方式维护日历月与月亮朔望周期的关系。在该方式中日历月的平均长度等于月亮朔望周期。在这种情况下,采用在特定年增加一个月的置闰方法将日历年调整到同地球围绕太阳公转的周期基本吻合。其他的日历,例如公历其设计并不与月亮朔望周期保持协调关系:日历年简单地分为整数月。

- d) 许多日历包括短于一个月的若干天的时间间隔。例如:公历和其他西方日历用一种七日星期作为标准间隔。虽然星期中的日是日期的文化意义元素,它没有增加日的时位置信息。由于某节日落在某星期的指定日以及特殊日子的需要,置闰规则是复杂的。
- e) 每个日历与一个或多个日历时代相关联。一个日历时代是基于与某些神话或历史事件对应的参考日期起算的年的序列。日历趋向于分为两类。有些日历相对于一个参考事件纪年,因此,年只落入一个或两个日历时代。公历就是这样的例子。由于多种原因,参考事件可能从一个历史时段变到另一个历史时段,例如儒略历(D. 3. 1)就使用了几个不同的参考事件。其他日历使用多个参考事件。以统治者在位纪年的帝王纪年方法是相当普遍的。现代日本历(D. 3. 2)和古代巴比伦历(D. 3. 3)都是实例。

D. 2 历法描述

本标准 5.3.2 描述为日历定义一组相对简单接口的模式。它规定描述日历时代的详细要求。支持日历日期到儒略日期转换的基本信息是由 D. 1 中讨论的相关因素决定的。

有些日历,包括正在使用的大多数日历已经规则化到这样一种程度,即已经设计出将一个日期精确地转换到儒略日期的算法。在这种情况下,所需要的信息可以算法的形式提供。Dershowitz 和 Reingold (1997), Doggett (1992), Hatcher (1984, 1985) 和 Richards (1998) 提供了大量这类算法。

许多日历几乎是规则的,可以推导出将日期转换成近似儒略日期值的算法。在这种情况下,所需要的信息可以算法的形式提供。它应包括对每个算法结果准确度的估计和可获得这种准确度的时段的陈述。

非常不规则的日历不能完全用算法描述。有时提供带有每年使用模式对照表的年度模式的描述是可能的。最可行的是提供对带有一系列参照日期的典型历年描述,它允许通过内插将日期转换成近似的儒略日期。Parise (1982) 提供了许多日历转换表。

D. 3 示例

D. 3. 1 儒略历

儒略历是规则日历的一个实例。它从公元前 45 年开始记日,当恺撒(Julius Caesar)为了使日历回到与回归年同步,进行古代罗马历法改革的时候。儒略历平年有 365 天,每 4 年插入一个闰日,该年被作为闰年。恺撒死后,罗马大祭司曲解了该规则,每 3 年增加一个闰日。在公元前 9 年,奥古斯都(Augustus)命令改正了由于疏忽置闰导致的错误,成功置闰直至公元 8 年。

儒略历系统已经被用于大量的日历时代。详见 Parise (1982) 和 Richards (1998)。在经过罗马帝

国年代的后来共和时期,纪年从推测的建立该城的日期即相当于公元前 753 年起算。在公元 525 年, Dionysius Exiguus 提议纪年从所谓的耶稣诞辰日起算。可是这种基督时代在西欧直到公元 11 世纪才被广泛应用,在希腊直到 15 世纪才被采用。

表 D.1 儒略年各月

月 名	平年长度	闰年长度
1 月	31	31
2 月	28	29
3 月	31	31
4 月	30	30
5 月	31	31
6 月	30	30
7 月	31	31
8 月	31	31
9 月	30	30
10 月	31	31
11 月	30	30
12 月	31	31

自奥古斯都以来,儒略年由 12 个月构成,如表 D.1 所示。在罗马帝国 1 月 1 日是该年的第一日。后来,使用其他日期。详见 Parise (1982)。

表 D.2 是满足 5.3 要求的儒略历描述的实例。该实例描述了与基督时代(约公元 1000 年 1 月 1 日至公元 1582 年 10 月 4 日,用 1 月 1 日作为该年第一天)一起使用的儒略历。

表 D.2 具有基督时代的儒略历的描述

元 素	值
TM_ReferenceSystem	
TM_ReferenceSystem.name	儒略历
TM_ReferenceSystem.domainOfValidity	西欧
TM_CalendarDateTrans 儒略历日期(Y/M/D)转换到儒略日(J)的算法	$Y' = Y + 4716 - (14 - M) / 12$ $M' = \text{MOD}(M + 9, 12)$ $D' = D - 1$ $c = (1461Y') / 4$ $d = (153M' + 2) / 5$ $J = c + d + D' - 1401$
TM_Calendar.julTrans 儒略日(J)转换到儒略历日期(Y/M/D)的算法	$J' = J + 1401$ $Y' = (4J' + 3) / 1461$ $T' = \text{MOD}(4J' + 3, 1461) / 4$ $M' = (5T' + 2) / 153$ $D' = \text{MOD}(5T' + 2, 153) / 5$ $D = D' + 1$ $M = \text{MOD}(M' + 2, 12) + 1$ $Y = Y' - 4716 + (14 - M) / 12$

表 D.2 (续)

元 素	值
基础	
TM_CalendarEra.name	基督时代
TM_CalendarEra.referenceEvent	耶稣基督诞辰日
TM_CalendarEra.referenceDate	01, 01, 01
TM_CalendarEra.julianReference	1 721 423.25
TM_CalendarEra.epochOfUse.begin	2 087 769
TM_CalendarEra.epochOfUse.end	2 299 160

在表 D.2 的算法已经被 Richards (1998) 改编。它们需要被更改是为了支持儒略历日期关联其他日历时代或不是以 1 月 1 日为 1 年的第一天的儒略历日期的转换。详见 Richards (1998)。所有的计算是整数算法的。

D.3.2 中国农历

中国采用的一种传统历法,这种历法安排一年中的年、月、日历日以及二十四节气等要素,供民众日常生活使用,特别用以指导农事活动,称为农历,又名夏历、旧历、中历,民间也有称阴历的。它用月亮的朔望¹⁾周期来定月份的长度,又用设置闰月的方法使年的平均长度与回归年长度(约 365.2422 日)相近,兼有阴历月和阳历年的性质,因此在实质上是一种阴阳合历。农历把日月合朔(太阳和月亮的黄经相等)的日期作为月首,即初一。朔望月约长 29.530 59 日,有的月份是 30 日,称大月;有的月份是 29 日,称小月。基本上取大、小月相间,有时加进连大月或连小月。月首所在的日期,要根据太阳和月球的位置推算确定,并非人为安排。农历以 12 个月为一年,共 354 或 355 日,与回归年相差 11 日左右,所以大约隔三年就要安插一个闰月,再过两年又安插一个闰月,平均约 19 年有 7 个闰月。安插闰月,和太阳在黄道上的位置有密切关系的 24 节气相关。在 24 节气中,冬至、大寒、雨水、立春等又称为 12 中气,而农历月份的名称同中气有密切的关系,如含“雨水”的月份叫正月,含“春分”的月份叫二月等等。如果某一月份不含中气,该月份就定为闰月,并用上个月的月份名,称闰某月。用这种办法置闰月可以使历月份与节气大致匹配,较好地调节了朔望月与回归年的关系。中国农历还采用干支法来纪年与纪日。

中国已于公元 1912 年开始采用公历,表 D.3 解释了公元 1990 年至 2009 年中国农历春节,用于描述由本标准规定的日历时代元素的应用。

表 D.3 近 20 年中国农历描述

名称	参考事件	参考日期	公历	儒略参考日期	时代使用时段	
					开始	结束
2009 年	春节	正月初一辛未	1 月 26 日	2 454 857.333	2 454 857.333	2 455 241.333
2008 年	春节	正月初一丁丑	2 月 07 日	2 454 503.333	2 454 503.333	2 454 857.333
2007 年	春节	正月初一癸未	2 月 18 日	2 454 149.333	2 454 149.333	2 454 503.333
2006 年	春节	正月初一戊午	1 月 19 日	2 453 764.333	2 453 764.333	2 454 149.333
2005 年	春节	正月初一甲子	2 月 09 日	2 453 410.333	2 453 410.333	2 453 764.333
2004 年	春节	正月初一庚子	1 月 22 日	2 453 026.333	2 453 026.333	2 453 410.333

- 1) 朔望:朔是指月球与太阳的地心黄经相同的时刻。这时月球处于太阳与地球之间,几乎和太阳同起同落,朝向地球的一面因为照不到太阳光,所以从地球上看不见。望是指月球与太阳的地心黄经相差 180° 的时刻。这时地球处于太阳与月球之间。月球朝向地球的一面照满太阳光,所以从地球上看来,月球呈光亮的圆形,称为满月或望月。

表 D.3 (续)

名称	参考事件	参考日期	公历	儒略参考日期	时代使用时段	
					开始	结束
2003 年	春节	正月初一乙巳	2 月 01 日	2 452 671.333	2 452 671.333	2 453 026.333
2002 年	春节	正月初一辛亥	2 月 12 日	2 452 317.333	2 452 317.333	2 452 671.333
2001 年	春节	正月初一丁亥	1 月 24 日	2 451 933.333	2 451 933.333	2 452 317.333
2000 年	春节	正月初一癸巳	2 月 05 日	2 451 579.333	2 451 579.333	2 451 933.333
1999 年	春节	正月初一己亥	2 月 16 日	2 451 225.333	2 451 225.333	2 451 579.333
1998 年	春节	正月初一乙亥	1 月 28 日	2 450 841.333	2 450 841.333	2 451 225.333
1997 年	春节	正月初一庚辰	2 月 07 日	2 450 486.333	2 450 486.333	2 450 841.333
1996 年	春节	正月初一丙戌	2 月 19 日	2 450 132.333	2 450 132.333	2 450 486.333
1995 年	春节	正月初一壬戌	1 月 31 日	2 449 748.333	2 449 748.333	2 450 132.333
1994 年	春节	正月初一丁卯	2 月 10 日	2 449 393.333	2 449 393.333	2 449 748.333
1993 年	春节	正月初一甲辰	1 月 23 日	2 449 010.333	2 449 010.333	2 449 393.333
1992 年	春节	正月初一庚戌	2 月 04 日	2 448 656.333	2 448 656.333	2 449 010.333
1991 年	春节	正月初一丙辰	2 月 15 日	2 448 302.333	2 448 302.333	2 448 656.333
1990 年	春节	正月初一壬辰	1 月 27 日	2 447 918.333	2 447 918.333	2 448 302.333

D.3.3 近代日本历

自 1873 年以来,除被维持的根据天皇时代纪年系统外,日本已经使用公历。表 D.4 解释了用于描述日历年代的由本标准规定的元素的应用。

表 D.4 日本日历年代

名称	参考事件	参考日期	儒略参考日期	时代使用时段	
				开始	结束
明治 Meiji	采用公历	明治 6.1.1	2 405 160	2 405 160	2 419 614
大正 Taisho	新天皇登基	大正 1.7.31	2 419 615	2 419 615	2 424 875
昭和 Showa	新天皇登基	昭和 1.12.26	2 424 876	2 424 876	2 447 534
平成 Heisei	新天皇登基	平成 1.1.8	2 447 535	2 447 535	至今

D.3.4 古代巴比伦历

古代巴比伦历是非常不规则日历的示例。它是在典型年中有 12 个月的月亮历(表 D.5)。为保持日历与季节的协调,置闰月为第 13 月。最初(约公元前 2400 年)是为了保证大麦收获季节发生在前 1 年的第 1 月的需要。到公元前 1000 年,根据月亮接近于昴星团为该年的第 1 天设计置闰。到公元前 500 年,置闰为 19 年一个循环,在第 3、6、8、11、14 和 19 年置闰 12 月(Adaru II)在第 17 年置闰 6 月(Ululu II)。

表 D.5 巴比伦年的月

月数	名称	月数	名称
1	Nisanu	7	Tashritu
2	Ayaru	8	Arakhsamnu
3	Simanu	9	Kislimu
4	Du'uzu	10	Tebetu
5	Abu	11	Shabatu
6	Ululu	12	Adaru
(6)	Ululu II	(12)	Adaru II

一天从日落开始,通常,月开始于最初看到新月时,因此月的长度是 29 日与 30 日不规则地交替。但是,如果新月未被观测到,该月仍结束于第 30 日。

注 1: 在大多数古代日历中,一日开始于日出或日落,因为它们是在每日循环中仅有的不用特殊设备帮助就可以观测的事件。

注 2: 原则上,用现代天文数据计算朔望月的长度是可能的,但是,由于历日月的长度是基于观测,它们可能不能与天文月相完全协调。

纪年的基础是帝王年代。年代开始于帝王登基后的第一年。在公元 2 世纪,托勒密提出了回溯到公元前 747 年的帝王及他们日期的列表。该列表被用作关联巴比伦年和更近代时代年的基础。还有其他的古代列表,该表可以将日期回溯到大约公元前 2000 年,虽然在早期的年精度较低。表 D.6 提供了巴比伦日历 10 年的示例。

表 D.6 巴比伦 Nebuchadnezzar II 在位首 10 年

年	儒略日期	月数	日数	闰月
Nebuchadnezzar 1	1500163.25	13	384	Adaru II
Nebuchadnezzar 2	1500547.25	12	355	
Nebuchadnezzar 3	1500902.25	12	356	
Nebuchadnezzar 4	1501258.25	13	382	Ululu II
Nebuchadnezzar 5	1501640.25	12	355	
Nebuchadnezzar 6	1501995.25	12	354	
Nebuchadnezzar 7	1502349.25	13	383	Adaru II
Nebuchadnezzar 8	1502732.25	12	355	
Nebuchadnezzar 9	1503085.25	13	383	Adaru II
Nebuchadnezzar 10	1503470.25	12	355	

由于特定的月的长度是未知的,执行这个日历日期与儒略日期的精确转换是不可能的。然而,一个基于任一给定年的月平均长度的算法可返回一个正确的儒略日期的接近值。表 D.7 是由本标准规定的用于描述古巴比伦历的元素使用的示例。

表 D.7 Nebuchadnezzar II 时间巴比伦历的描述

元 素	值
TM_ReferenceSystem	
TM_ReferenceSystem.name	巴比伦历
TM_ReferenceSystem.domainOfValidity	西南亚
TM_Calendar.dateTrans 从巴比伦历日期(Y/M/D)转换到儒略日(J)的算法	$J = J_Y + \text{INT}((M-1)(L/N)) + D - 1$ 其中: J_Y — Y年尼撒月第一天的儒略日数(表中); M —月份数; L —该年的天数(表中); N —该年的月数(表中); D —该天是该月中的第几天数。
TM_Calendar.julTrans 从儒略日(J)转换到巴比伦历日期的算法	$Y = \text{表中尼撒月第一天的儒略日数}(J_Y) \text{ 小于 } J \text{ 的最大值对应的年}$ $M = \text{INT}((J - J_Y + 1)/(L/N))$ $D = J - J_Y + 1 - \text{INT}(M * (L/M))$
基础	
TM_CalendarEra.name	尼布甲尼撒二世
TM_CalendarEra.referenceEvent	尼布甲尼撒二世登上巴比伦王国王位
TM_CalendarEra.referenceDate	01,01,01
TM_CalendarEra.julianReference	1721423.25
TM_CalendarEra.epochOfUse.begin	2087769
TM_CalendarEra.epochOfUse.end	2299160

D.3.5 全球定位系统历

全球定位系统(GPS)使用特殊的用于描述GPS有关数据的时间位置的日历和时钟。日期由周数(WN)确定,以开始于1980年1月6日午夜(00:00:00 UTC)的日历时代计算,日由在周中的顺序日数(DN)确定。两种情况(周数和日数)记数都从0开始。表D.8是由本标准规定的用于描述GPS历的要素的使用示例。

表 D.8 GPS 历描述

元 素	值
TM_ReferenceSystem	
TM_ReferenceSystem.name	全球定位系统历
TM_ReferenceSystem.domainOfValidity	全球
TM_Calendar.dateTrans 转换日历日期(WN, DN)到儒略日期(J)的算法	$J = 2\,444\,244.5 + (7 * WN) + DN$
TM_Calendar.julTrans 转换儒略日期(J)到日历日期(WN, DN)的算法	$DL = J - 2\,444\,244.5$ $WN = \text{INT}(DL/7)$ $DN = \text{MOD}(DL, 7)$
基础	
TM_CalendarEra.name	GPS时代
TM_CalendarEra.referenceEvent	
TM_CalendarEra.referenceDate	0001,01
TM_CalendarEra.julianReference	244 424 4.5
TM_CalendarEra.epochOfUse.begin	244 424 4.5
TM_CalendarEra.epochOfUse.end	当前时刻

GPS 时间系统是与众不同的,它以周的时间(TOW)而不是以日的时间规定表示秒的时钟时间。GPS 一周包括 604 800s。GPS 时间与 UTC 不同有两个理由:GPS 时间是连续标度的,而 UTC 是用增加闰秒与地球自转时间进行周期性调整的;相对于 UTC,GPS 时间是以一微秒级变化的。因此,GPS 时间和 UTC 之间的转换算法是相当复杂的,且未被包括在本例中。

附录 E

(资料性附录)

本标准中的类名中英文对照表

表 E.1 类名中英文对照表

英文名称	中文名称
TM_Object	TM_对象
TM_Primitive	TM_单形
TM_GeometricPrimitive	TM_几何单形
TM_TopologicalPrimitive	TM_拓扑单形
TM_Complex	TM_复形
TM_TopologicalComplex	TM_拓扑复形
TM_Instant	TM_时刻
TM_Period	TM_时段
TM_Order	TM_顺序
TM_Separation	TM_间隔
TM_TemporalPosition	TM_时间位置
TM_Position	TM_位置
TM_RelativePosition	TM_相对位置
TM_Duration	TM_持续时间
TM_OrdinalReferenceSystem	TM_顺序参照系
TM_ReferenceSystem	TM_参照系
TM_PeriodDuration	TM_时段持续时间
TM_IntervalLength	TM_间隔长度
TM_Node	TM_结点
TM_Edge	TM_边
TM_Calendar	TM_日历
TM_Clock	TM_时钟
TM_CalendarEra	TM_日历时代
TM_CallDate	TM_日历日期
TM_CoordinateSystem	TM_坐标系
TM_OrdinalEras	TM_顺序时代
TM_IndeterminateValue	TM_不确定值
TM_DateAndTime	TM_日期和时间
TM_OrdinalPosition	TM_时间顺序位置
GF_TemporalAttributeType	GF_时间属性类型
GF_FeatureType	GF_要素类型
TM_Event	TM_事件
TM_State	TM_状态

参 考 文 献

- [1] ALLEN J F. *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM, 1983, 26: 832-843.
- [2] N DERSHOWITZ, E M REINGOLD. *Calendrical calculations*. Cambridge University Press, 1997.
- [3] L E DOGGETT. *Calendars* in P. K. SEIDELMANN (editor). Explanatory supplement to the astronomical almanac and the American ephemeris and nautical almanac, University Science Books, Sausalito, CA, USA, 1992: 575-608.
- [4] D A HATCHER. *Simple formulae for Julian day numbers and calendar dates*. Journal of the Royal Astronomical Society, 1984, 25: 53.
- [5] D A HATCHER. *Generalized equations for Julian day numbers and calendar dates*. Journal of the Royal Astronomical Society, 1985, 26: 151.
- [6] International Telecommunications Union, ITU-R Recommendation TF. 686-1 (10/97), Glossary 1997.
- [7] International Telecommunications Union, ITU-R Recommendation TF. 460-5 (10/97), Standard-frequency and time-signal emissions, 1997.
- [8] ISO 31-2:1992 Quantities and units—Part 2: Periodic and related phenomena.
- [9] ISO 19104²⁾ Geographic information—Terminology.
- [10] C. S. JENSEN, et al. *A consensus glossary of temporal data base concepts*, ACM SIGMOD Records 1994, Vol. 23 Also available as consGlos. ps from ftp://ftp.cs.arizona.edu/tsql/doc/.
- [11] Object Management Group. *OMG Unified Modeling Language Specification, version 1.3*, 1999, Available from World Wide Web at <http://www.omg.org/cgi-bin/doc?ad/99-06-08>.
- [12] F PARISE. *The book of calendars*. Facts on file. New York 1982.
- [13] J P PARISOT. *Additif to the paper of H. A. Hatcher*. Journal of the Royal Astronomical Society, 1986, 27: 506.
- [14] E G RICHARDS. *Mapping Time: The calendar and its history*. Oxford University Press, 1998.
- [15] 中国大百科全书出版社编辑部, 中国大百科全书总编辑委员会《天文学》编辑委员会. 中国大百科全书: 天文学[M/CD]. 1.1 版. 中国大百科全书出版社.

2) 即将出版。

中 华 人 民 共 和 国
国 家 标 准
地 理 信 息 时 间 模 式

GB/T 22022—2008/ISO 19108:2002

*

中国标准出版社出版发行
北京复兴门外三里河北街16号
邮政编码:100045

网址 www.spc.net.cn

电话:68523946 68517548

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 3.25 字数 89 千字

2008年9月第一版 2009年6月第三次印刷

*

书号: 155066·1-33191 定价 34.00 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)68533533