JSGI2.0

地理情報標準第２版（JSGI2.0）
II　時間スキーマ

2002 年 3 月

ワーキンググループ１　作成

目　次

序文 ····· 7
概要 ····· 7
1．適用範囲 ····· 7
2．適合性 ····· 8
2.1 適合性クラスと要件 ····· 8
2.2 応用スキーマ ····· 8
2.2.1 データ交換のための応用スキーマ ····· 8
2.2.2 操作を含むデータの応用スキーマ ····· 8
2.3 地物カタログ ····· 8
2.4 メタデータ要素の仕様 ····· 8
2.5 データ集合のためのメタデータ ····· 8
3．引用規格 ····· 8
4．定義 ····· 8
5．地理情報の時間に関する概念スキーマ ····· 11
5.1 スキーマの構造 ····· 11
5.2 時間の幾何 ····· 12
5.2.1 次元としての時間 ····· 12
5.2.2 時間のオブジェクト ····· 12
5.2.3 時間幾何プリミティブ ····· 13
5.2.4 時間位相オブジェクト ····· 18
5.3 時間参照系 ····· 21
5.3.1 時間参照系の型 ····· 21
5.3.2 暦と時法 ····· 22
5.3.3 時間座標系 ····· 24
5.3.4 順序時間参照系 ····· 25
5.4 時間位置 ····· 26
5.4.1 序 ····· 26
5.4.2 グレゴリオ暦と UTC で参照される時間位置 ····· 27
5.4.3 TM_Position（時間位置情報データ型） ····· 27
5.4.4 暦と時法で参照される時間位置 ····· 28
5.4.5 時間座標系で参照される時間位置 ····· 29
5.4.6 順序時間参照系で参照される時間位置 ····· 29
5.5 時間と地理情報の構成要素 ····· 29
5.5.1 地理情報構成要素の時間的な性質 ····· 29
5.5.2 地物の時間属性 ····· 30
5.5.3 時間地物操作 ····· 32
5.5.4 時間と地物関係 ····· 32
5.5.5 時間メタデータ要素 ····· 35

附属書 A （規定）　抽象試験項目群 ･･37
A.1 応用スキーマ ･･37
A.1.1 データ交換のための応用スキーマ ･･37
A.1.2 操作を伴うデータに関する応用スキーマ ･･････････････････････････････････････37
A.2 地物カタログ ･･37
A.3 メタデータ要素の仕様 ･･38
A.4 データ集合のためのメタデータ ･･38
附属書 B （参考）　応用スキーマにおける時間の使用 ･･････････････････････････････39
B.1 時間地物属性 ･･39
B.1.1 データ型としての TM_GeometricPrimitive（時間幾何プリミティブ） ･･････････････39
B.1.2 時間属性としての TM_GeometricPrimitive ･･････････････････････････････････39
B.1.3 属性としての TM_TopologicalComplex（時間位相複体） ･･････････････････････････40
B.1.4 循環性のある属性値 ･･40
B.2 時間地物関係 ･･40
B.2.1 単純な時間関係 ･･40
B.2.2 地物継起 ･･42
B.3 時間特性をもった地物関係 ･･43
附属書 C（規定） メタデータにおける時間参照系の記述について ･･････････････････････44
C.1 時間参照系に関するメタデータ ･･44
附属書 D （参考）　暦の記述 ･･48
D.1 暦の内部構造 ･･48
D.2 暦の記述 ･･49
D.3 実例 ･･50
D.3.1 ユリウス暦 ･･50
D.3.2 現代日本の和暦 ･･52
D.3.3 古代バビロニア暦 ･･52
D.3.4 GPS 暦 ･･54
附属書 E（参考）　参考文献 ･･56

図

図 5.1 － 時間スキーマの構造 ･･ 11
図 5.2 － 時間オブジェクト ･･ 12
図 5.3 － 時間幾何プリミティブ ･･ 13
図 5.4 － 相対時間位置型（TM_RelativePosition） ･･････････････････････････････ 15
図 5.5 － 時間長型 ･･ 17
図 5.6 － 時間の位相 ･･ 19
図 5.7 － 時間参照系 ･･ 22
図 5.8 － 暦と時法 ･･ 23
図 5.9 － 時間座標系 ･･ 25
図 5.10 － 順序時間参照系 ･･ 26
図 5.11 － 時間位置に関するデータ型 ･･ 28

図 5.12 － 一般地物モデルのメタクラス …… 30
図 5.13 － 地物の時間属性型 …… 31
図 5.14 － 時間地物関係の分類 …… 34
図 5.15 － 拡張されたメタデータ要素 …… 36
図 B.1 － データ型としての TM_GeometricPrimitive …… 39
図 B.2 － 時間地物属性としての TM_GeometricPrimitives …… 39
図 B.3 － 時間地物属性としての TM_TopologicalComplex …… 40
図 B.4 － 周期性のある属性値 …… 40
図 B.5 － 単純な時間関係の明示的表現 …… 41
図 B.6 － Building スキーマ …… 41
図 B.7 － 地物型間での地物継起 …… 42
図 B.8 － 汎化した地物レベルでの地物継起 …… 43
図 B.9 － 放牧地の賃貸 …… 43
図 D.1 － 典型的な暦の内部構造 …… 48

表

表 B.1 － Building スキーマのデータ集合 …… 41
表 B.2 － Building インスタンス間の単純な時間関係 …… 41
表 C.1 － 時間参照系を記述するためのメタデータ要素 …… 44
表 D.1 － ユリウス年の各月 …… 50
表 D.2 － キリスト年代によるユリウス暦の記述 …… 51
表 D.3 － 日本の暦年代 …… 52
表 D.4 － バビロニア年の各月 …… 52
表 D.5 － バビロニアにおけるネブチャドネッツァ 2 世の治世の最初の 10 年 …… 53
表 D.6 － ネブチャドネッツァ 2 世の時代からのバビロニア暦についての記述 …… 53
表 D.7 － GPS 暦の記述 …… 54

序文　地理情報標準第 2 版（JSGI2.0）のこの標準は，国土地理院及び民間企業 38 社との共同研究において，1999 年 3 月に作成した地理情報標準第 1 版から，ワーキンググループ 1 が改良して作成したものである。

この標準で，点線の下線を施してある箇所は，原国際規格原案（ISO/DIS 19108 Geographic information — Temporal schema）にない事項である。

概要　この標準は，実世界から抽象化された地理情報の時間特性を記述するために必要となる標準的な概念を定義する。地理情報の時間特性には，時間の領域における値をもった地物属性，地物操作，地物関係及びメタデータ要素が含まれる。

コンピュータと地理情報システムの広範な利用は，多様な分野における地理空間データ解析の増加をもたらした。地理情報は三次元空間領域に限定されるものではなく，多くの地理情報システムは時間的な特性をもったデータを必要としている。時間特性に関して標準化された概念スキーマを，シミュレーション及び予測モデリングなどの応用システムで利用することによって，地理情報の能力が増大するのである。

基本的な物理的現実として，すべての科学的及び技術的分野が時間に関心をもっている。この標準で記述される概念の多くは地理情報以外の分野にも適用できる。この標準は時間の記述をおこなうための独立した標準を規定するのではなく，地理データ集合及び地物の時間特性の記述法を標準化することを意図している。地理情報システム及びソフトウェア開発者，地理情報の利用者は，広く理解が得られるような時間データ構造としてこのスキーマを使うことになる。

歴史的には，地物の時間特性は地物の主題属性として扱われてきた。例えば，地物“建物”は“建築日付”を属性としてもつことができる。しかしながら，地物の振る舞いを時間の関数として記述したいという関心が高まってきている。これについては，時間を空間と独立なものとして取り扱うことによって多少の対応は可能である。例えば移動するオブジェクトの経路は，ある時点でのオブジェクトの空間的な位置を表現し，その位置を通過した時間情報を属性としてもっている“経過点”という地物の集合として表現することができる。時間に関連する振る舞いは，もし時間次元が空間次元と結びついて，地物を時空間オブジェクトとして表現できるようなれば，より簡単に表現することが可能になる。例えば，移動するオブジェクトの経路は，座標 x，y 及び t によって記述される曲線として表現できるであろう。この標準は地物属性としての時間の取り扱い方を標準化するために用意されたものである。これは，地物の幾何属性を空間と時間の座標の組合せによって記述するものではないが，将来の標準によってそのことがなされるための基礎を確立するように書かれている。

1．適用範囲　この標準は地理情報の時間特性を記述するための概念を定義する。この標準は時間情報を交換するために規定された既存の情報技術標準に基くものである。この標準は時間に関する地物属性，地物操作及び地物関係を定義し，さらに地理情報に関するメタデータに対し，時間に関する情報を定義するための基盤を提供する。この標準は，現実世界を抽象化した地理情報における時間特性にかかわったものであるため，トランザクション時間より，有効時間を重要視している。

2.　適合性

2.1　適合性クラスと要件　この標準では，試験項目の特性に応じて，四つの適合性クラスの箇条を定義している。

2.2　応用スキーマ

2.2.1　データ交換のための応用スキーマ　この標準に適合するためには，応用スキーマは，附属書 A の抽象試験項目群の A.1.1 の要件を満たさなければならない。

2.2.2　操作を含むデータの応用スキーマ　この標準に適合するためには，応用スキーマは，附属書 A の抽象試験項目群の A.1.2 の要件を満たさなければならない。

2.3　地物カタログ　この標準に適合するためには，応用スキーマは，附属書 A の抽象試験項目群の A.2 の要件を満たさなければならない。

2.4　メタデータ要素の仕様　この標準に適合するためには，応用スキーマは，附属書 A の抽象試験項目群の A.3 の要件を満たさなければならない。

2.5　データ集合のためのメタデータ　この標準に適合するためには，応用スキーマは，附属書 A の抽象試験項目群の A.4 の要件を満たさなければならない。

3.　引用規格　次に掲げる規格は，この標準に引用されることによって，この規定の一部を構成する。これらの引用規格のうちで発行年を付記してあるものは，記載の年の版だけがこの標準の規定を構成するものであって，その後の改正版・追補には適用しない。発行年を付記していない引用規格は，その最新版（追補を含む）を適用する。

地理情報標準　応用スキーマのための規則

地理情報標準　地物カタログ化法

地理情報標準　メタデータ

ISO TS/DTS 19103＊　Geographic information — Conceptual schema language

ISO 31-1:1992　量と単位 — パート 1: 空間と時間

ISO 1000:1992　SI 単位及びその複合的利用及びその他の単位の併用に関する勧告

ISO 8601:1988 and Corrigendum 1:1991　データ要素と交換フォーマット－情報交換－日付と時間の表現

ISO 11404:1996　情報技術－プログラム言語，その環境とシステムソフトウェア・インタフェース－言語に独立なデータ型

注＊　未発行

4.　定義　この標準で用いる主な定義は，次による。

4.1　暦（calendar）　離散的な時間参照系で，一日をその分解能とする時間位置を定義するための根拠を与える

4.2　暦年代（calendar era）　指定された日付からのある期間の列として構成された一つの暦の分割

4.3　暦年（calendar year）　太陽の周りを地球が一回転する周期にほぼ等しい長さの期間

備考　暦年の長さは整数で表す日数で示される。暦年の日数はそれぞれの暦の規則によって異なる。

4.4　協定世界時（Coordinated Universal Time）　UTC　国際度量衡協会及び標準周期と時

間シグナルの等位通報の基盤を形成する国際地球回転局によって維持されている時間の目盛

4.5　循環（cycle）　事象又は状態の規則的な繰り返し

4.6　日（day）　地球の自転周期にほぼ等しい長さをもつ期間

4.7　時間長（duration）　時間の量

備考　時間長は，時間の長さと等しい。

4.8　エッジ（edge）　一次元の位相プリミティブ

備考　エッジの幾何学的実現は曲線である。エッジの境界は位相複合体中でこのエッジと関連付けられた一つ又は二つのノードの集合である。

4.9　事象（event）　ある瞬間に起こる動作

4.10　地物（feature）　実世界の現象の抽象概念

備考　地物は型又はインスタンスとして現れる。地物型又は地物インスタンスという用語は，一方のみを意味するときに使われるべきである。

4.11　地物属性（feature attribute）　地物の特性

備考　地物属性型は名前，データ型及びそれに関連する値の定義域を有する。地物属性インスタンスは地物属性型の定義域から得られる属性値を有する。

4.12　地物分裂（feature division）　直前に存在した地物が，同じ型をとる二つ以上の別の地物インスタンスに置き換わる地物継起

例　地物型“土地区画”のあるインスタンスが，法律的に分割されて同じ地物型の二つのインスタンスに置き換えられる。

4.13　地物融合（feature fusion）　直前に存在した，同じ地物型の二つ以上のインスタンスが，同じ地物型の一つのインスタンスに置き換わる地物継起

例　地物型“牧場”の二つのインスタンスは，各々を隔てるフェンスが取り除かれたときに一つのインスタンスに置き換わる。

4.14　地物操作（feature operation）　ある地物型のすべてのインスタンスが実行できる又は影響を受ける操作

例　地物型“堰”上の地物操作は水門を揚げる。この操作の結果は水門の高さを揚げ，貯水池の水位を上げる。

備考　地物操作は地物型の定義の根拠を提供する。

4.15　地物関係（feature relationship）　一つの地物型のインスタンスを，同じ又は異なる地物型のインスタンスと関連付けるような関係

4.16　地物置換（feature substitution）　一つの地物インスタンスが同じ又は異なる型の別の地物インスタンスへ置き換わる地物継起

例　地物型“建物”のインスタンスが取り壊されて，地物型“駐車場”のインスタンスに置き換わる。

4.17　地物継起（feature succession）　一つ以上の地物インスタンスが存在しなくなって，別の一つ以上の地物インスタンスに置き換わること

4.18　幾何プリミティブ（geometric primitive）　空間の単一の連結で均質な要素を表す幾何オブジェクト

備考　幾何プリミティブは幾何構成についての情報を表す不可分なオブジェクトである。これには点，曲線，曲面及び立体が含まれる。

4.19　グレゴリオ暦（Gregorian calendar）　ユリウス暦の誤差を修正するために，1582 年に最初に導入された，はん用的な暦

備考　グレゴリオ暦では，通常の年は 365 日，閏年は 366 日あり，それを 12 に分割した月の列から成る。

4.20　瞬間（instant）　時間における位置を表現する点

4.21　間隔目盛（interval scale）　値の間の順序付けと距離の両方を記述するために使用される任意の原点をもつ目盛

備考　ある間隔目盛で測定された二つの値の比は，意味をもたない。

4.22　ユリウス日（Julian date）　直前の正午からの経過を表す 10 進小数の日数を伴ったユリウス日数

4.23　ユリウス日数（Julian day number）　ユリウス暦における紀元前 4713 年 1 月 1 日のグリニジ標準時正午を原点として数えた日の数

4.24　存続期間（life span）　物が存在し続けている期間

備考　有効時間による存続期間は，物体がモデル化された現実において存在している期間を表す。トランザクション時間による存続期間は，データベース上の物体がデータベース内に存在している期間を表す。

4.25　月（month）　月の循環周期の長さとほぼ等しい期間

備考　ひと月の時間長は整数で表す日数である。ひと月の日数は個々の暦の規則で定められている。

4.26　ノード（node）　0 次元の位相プリミティブ

備考　ノードの境界は空集合である。

4.27　順序年代（ordinal era）　時間の中で順序付けられ名前が付けられた期間の集合のひとつ

4.28　順序時間参照系（ordinal temporal reference system）　順序年代の階層からなる時間参照系

4.29　順序目盛（ordinal scale）　オブジェクトの相対的な時間位置関係だけを測るときの基礎を提供する目盛

4.30　期間（period）　二つの異なる時間位置を境界にもつ時間の範囲を表現する一次元幾何プリミティブ

4.31　周期長（periodic time）　一周期の時間長

4.32　点（point）　位置を表現する 0 次元の幾何プリミティブ

4.33　状態（state）　ある期間をとおして持続するありさま

備考　個々の地物属性の名前や値は，その地物のありさまを記述している。

4.34　時間座標（temporal coordinate）　時間座標系の基礎として使用される間隔目盛の原点からの距離

4.35　時間座標系（temporal coordinate system）　単一の時間単位の倍数として距離が測定された間隔目盛を基礎とする時間参照系

4.36　時間地物操作（temporal feature operation）　時間の関数として定義された地物操作

4.37　時間地物関係（temporal feature relationship）　時間又は時間的な制約条件を参

照することによって特徴付けられる地物関係

4.38　時間位置（temporal position）　時間参照系に関係した位置

4.39　時間参照系（temporal reference system）　時間の計測に対応する参照系

4.40　時刻（time of day）　暦日中のある瞬間の時間位置の明示

4.41　位相複体（topological complex）　境界操作について閉じている位相プリミティブの集まり

備考　境界演算について閉じているとは，位相複体に含まれる位相プリミティブの境界オブジェクトもまたこの位相複体に含まれることを意味する。

4.42　位相プリミティブ（topological primitive）　単独の不可分な要素を表す位相オブジェクト

備考　位相プリミティブは幾何実現の同じ次元の幾何プリミティブの内部に相当する。

4.43　トランザクション時間（transaction time）　ある事実がデータベース内に存続し，検索可能である時間

4.44　有効時間（valid time）　ある事実が抽象化された現実の中で真である時間

5.　地理情報の時間に関する概念スキーマ

5.1　スキーマの構造　この箇条は，地理情報の時間に関する特性を記述する概念スキーマを示している。このスキーマは統一モデリング言語（UML）　［Object Management Group（1999）］　によって記述される。**地理情報標準－応用スキーマのための規則附属書 E** は，この標準全体で使われる UML の使用法について記述している。UML クラスには，属性，操作及び関連の三つの基本的な特性があり，このスキーマではそのすべてを使用する。このスキーマは抽象モデルであり，ある実装がこの標準に適合するためには，抽象モデルにおけるそれぞれの要素で記述された内容を実現する能力がなければならないが，同じ方法で実装する必要はない。

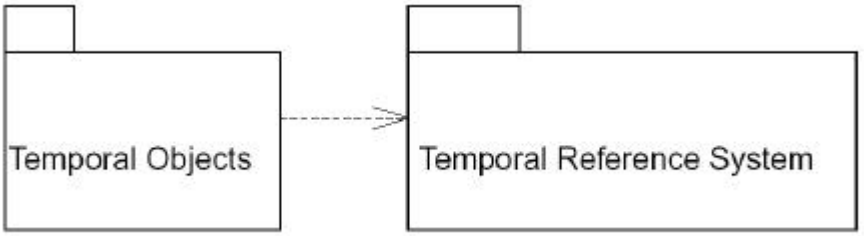

図 5.1 － 時間スキーマの構造

このスキーマは二つのパッケージから成る（図 5.1）。時間オブジェクト・パッケージ（5.2 参照）は時間幾何オブジェクト及び時間位相オブジェクトを定義しているが，これらは，地物とデータ集合の時間特性の値として使われなければならない。オブジェクトの時間位置は時間参照系と関係づけて指定されなければならない。時間参照系パッケージ（5.3，5.4）は時間参照系を記述するための要素を提供する。5.5 は，5.2 から 5.4 までの箇条で記述された概念が，地理情報の内容記述の中で，どのように使用されなければならないかを記述している。

この標準で定義している UML クラス名は，特定の標準を識別する二文字と一文字の下線か

ら成る接頭語で開始しており，パッケージ名も可能な限りこれに沿っている。TM_が，この標準で定義したクラスを識別する接頭語である。

5.2　時間の幾何

5.2.1　次元としての時間　時間は任意の空間次元に類似した一つの次元である。空間と同様，時間は幾何的及び位相的特性をもっている。時間における点は，時間参照系と関係づけて識別できる位置を占める。距離も測定できる。しかし，空間とは異なり，時間は単一次元である。すなわち，時間参照系は，いろいろな応用システムで空間的な位置を記述するために使用される線型の参照系と類似している。時間は絶対的な方向性をもち，時間における動きは常に前方向ではあるが，時間を測定する際には二つの方向で測ることができる。

備考　時間は，概念レベルでは常に幾何的及び位相的特性をもつものではあるが，時として，幾何的特性だけであったり又は位相的特性だけであると記述することは可能であり，また，それが望ましいことがある。

時間は二種類の目盛で測定される。順序目盛は時間に関する相対的な位置の情報だけを提供するが，間隔目盛は時間長を測定するための基礎を提供する。

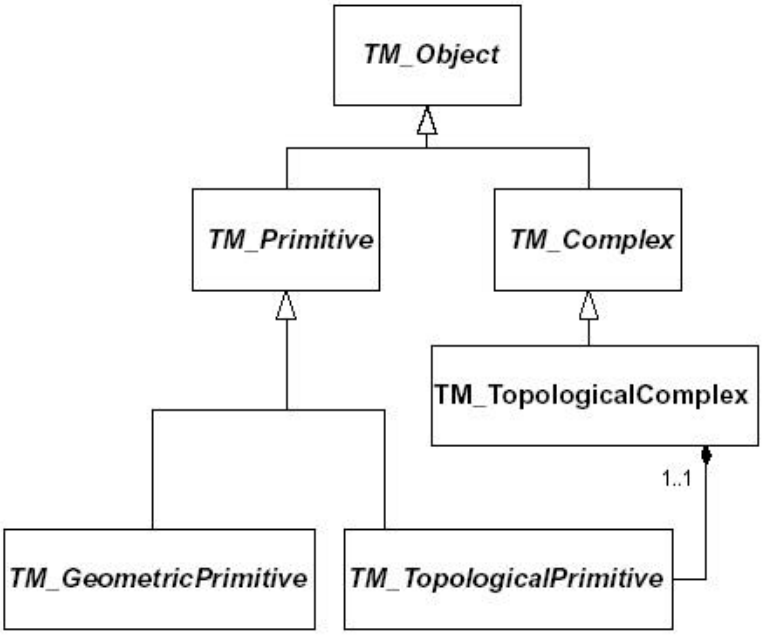

図 5.2 － 時間オブジェクト

5.2.2　時間のオブジェクト　時間幾何オブジェクト及び時間位相オブジェクトは，地物とデータ集合の時間特性を示す値として使用されなければならない。その解説ならびに例として 5.5 及び附属書 B を参照すること。TM_Object（時間オブジェクト要素）（図 5.2）は，二つの下位型をもつ抽象クラスである。TM_Primitive（時間プリミティブ）は，分解できない幾何又は位相要素を表現する抽象クラスである。TM_Primitive には二つの下位型がある。TM_GeometricPrimitive（時間幾何プリミティブ）（5.2.4.2）は時間位置に関する情報を提供する。TM_TopologicalPrimitive（時間位相プリミティブ）（5.2.4.2）は時間における接続性に関する情報を提供する。TM_Complex（時間複体要素）は複数の TM_Primitive の集約で

ある。TM_TopologicalComplex（時間位相複体要素）（5.2.4.5）は，この標準で定義された TM_Complex の唯一の下位型であり，接続関係にある複数の TM_TopologicalPrimitive の集約である。

5.2.3 時間幾何プリミティブ

5.2.3.1 時間幾何プリミティブ・クラス　時間軸の次元には，瞬間と期間の二つの幾何プリミティブが存在する。これらのプリミティブは間隔目盛で計測される時間の場合は解析的に定義され，順序目盛で計測される場合は類推的に定義される。TM_GeometricPrimitive（時間幾何プリミティブ）は二つの下位型 TM_Instant（瞬間要素）と TM_Period（期間要素）（図 5.3）をもつ抽象クラスである。TM_GeometricPrimitive は，TM_Order（時間順序インタフェース）に依存する TM_Primitive を継承し，TM_Separation（時間距離インタフェース）にも依存関係をもつ。依存関係に付加されたステレオタイプ“uses”は，そのクラスが，インタフェースとして定義された操作の中で，全部でなくてもよいがその幾つかが利用可能であることを意味している。

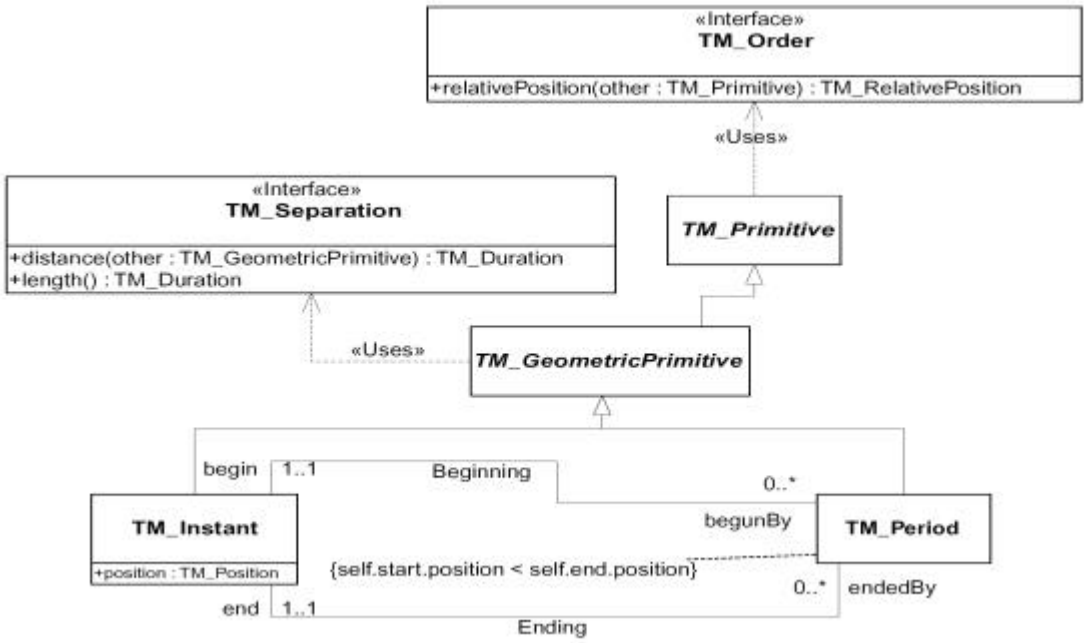

図 5.3 － 時間幾何プリミティブ

5.2.3.2 TM_Instant（瞬間要素）　瞬間は0次元幾何プリミティブであり，時間上の点を表現する。これは空間における点と同等である。実際には，瞬間は時間目盛の分解能よりも短い時間長を時間間隔としてもっている。

属性:　TM_Instant（瞬間要素）は一つの属性をもつ。

a) position:TM_Position（位置：時間位置情報データ型）は，この TM_Instant の位置を指定しなければならない。TM_Position（時間位置情報データ型）は，5.3 で規定される単一の時間参照系に関連づいていなければならない。TM_Instant のインスタンスは他と識別できるオブジェクトであるが，TM_Position のインスタンスはデータ値である。与えられた TM_Instant の TM_Position は，異なる時間参照系に関連づいた TM_Position によって置換することが可能である。

5.2.3.3　TM_Period（期間要素）　期間は時間の範囲を表現する一次元幾何プリミティブである。期間は空間における曲線と同等であり，曲線のように，始点及び終点（各瞬間）を境界にもつ開区間を成し，長さ（時間長）をもつ。その時間上の位置は，それが始まる瞬間及び終わる瞬間の時間位置によって記述され，その長さは二つの時間位置間の時間距離に等しい。

順序目盛では時間の長さを計測することが不可能なので，瞬間と期間の区別をすることはできない。実際には，単一の事象が起きた時間は，順序目盛で測られた瞬間と考えられる。連続した事象の列は時間の，ある一定の間隔を占めることとなるが，これがすなわち期間である。期間という用語は，一般的に他と区別される特性をもつ事象の列に対し，共通して適用される。

関連:

a)　Beginning（始点）は，期間開始時点を示す TM_Instant（瞬間要素）と，この TM_Period（期間要素）を関連付ける。

b)　Ending（終点）は，期間終了時点を示す TM_Instant と，この TM_Period を関連付ける。様々な理由から，期間の開始時点及び終了時点を表す TM_Instant の時間位置は不確定であり得る。不確定な時間位置に関する議論については 5.4.3 を参照せよ。

5.2.3.4　TM_Order（時間順序インタフェース）　TM_GeometricPrimitive（時間幾何プリミティブ）は，TM_Primitive（時間プリミティブ）の TM_Order（時間順序インタフェース）への依存関係を継承する。TM_Order は，自身の TM_Primitive の，他の TM_Primitive からの相対位置を判定する操作を提供する。

操作:

a)　relativePosition (other:TM_Primitive) : TM_RelativePosition
相対的な時間位置の取り得る値は，列挙データ型 TM_RelativePosition（相対時間位置型）（図 5.4）によって規定される。これは，Allen（1983）によって識別された 13 個の時間関連にもとづくものである。この操作は，この TM_GeometricPrimitive に対して，次に示す列挙データ型 TM_RelativePosition の値を戻さなければならない。

b)　この TM_Primitive と other で指定されたものが両方とも TM_Instant（瞬間要素）である場合，この操作は次に示す TM_RelativePosition の値を戻さなければならない。

戻り値	条件
Before	self.position<other.position
Equals	self.position=other.position
After	self.position>other.position

c)　この TM_Primitive が TM_Period（期間要素），other で指定されたものが TM_Instant である場合，この操作は次に示す TM_RelativePosition の値を戻さなければならない。

戻り値	条件
Before	self.end.position < other.position
EndedBy	self.end.position = other.position
Contains	self.begin.position <other.position AND self.end > other.position

BegunBy	self.begin.position = other.position
After	self.begin.position > other.position

d) この TM_Primitive が TM_Instant，other で指定されたものが TM_Period である場合，この操作は次に示す TM_RelativePosition の値を戻さなければならない。

戻り値	条件
Before	self.position<other.begin.position
Begins	self.position=other.begin.position
During	self.position> other.begin.position AND self.position<other.end.position
Ends	self.position=other.end.position
After	self.position> other.end.position

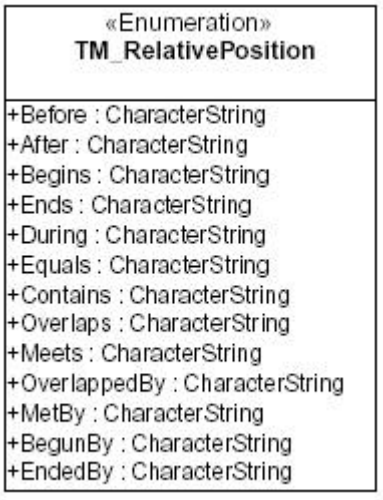

図 5.4 － 相対時間位置型（TM_RelativePosition）

e) この TM_Primitive と other で指定されたものが両方とも TM_Period である場合，この操作は次に示す TM_RelativePosition の値を戻さなければならない。

戻り値	条件
Before	self.end.position < other.begin.position
Meets	self.end.position = other.begin.position
Overlaps	self.begin.position < other.begin.position AND self.end.position > other.begin.position AND self.end.position < other.end.position
Begins	self.begin.position = other.begin.position AND self.end.position < other.end.position
BegunBy	self.begin.position = other.begin.position AND self.end.position > other.end.position
During	self.begin.position > other.begin.position AND self.end.position < other.end.position

Contains	self.begin.position < other.begin.position AND self.end.position > other.end.position
Equals	self.begin.position = other.begin.position AND self.end=other.end.position
OverlappedBy	self.begin.position > other.begin.position AND self.begin.position < other.end.position AND self.end.position > other.end.position
Ends	self.begin.position > other.begin.position AND self.end.position = other.end.position
EndedBy	self.begin.position < other.begin.position AND self.end.position = other.end.position
MetBy	self.begin.position = other.end.position
After	self.begin.position > other.end.position

もし，TM_Positionによる入力値のいずれかが不定である場合，この操作は例外条件を起動しなければならない。

5.2.3.5　TM_Separation（時間距離インタフェース）　TM_GeometricPrimitive（時間幾何プリミティブ）は，長さと距離を計算するための操作を提供するTM_Separation（時間距離インタフェース）との依存関係ももつ。TM_Duration（時間長データ型）（図5.5）はこれらの操作の戻り値となるデータ型である。

a)　length():TM_Durationは，このTM_GeometricPrimitiveの時間長を戻さなければならない。TM_Instant（瞬間要素）の時間長は定義により0である。TM_GeometricPrimitiveがTM_Period（期間要素）であるときは，TM_Period.begin及びTM_Period.endで指定される時間位置の間の距離を戻さなければならない。この操作は，いずれかのTM_Position（時間位置情報データ型）が不確定の場合は例外処理を起動する。両方のTM_Positionが，TM_OrdinalReferenceSystem（順序時間参照系）に関連している場合は，期間に長さが保持されているか，又はTM_Period.begin及びTM_Period.endが値を保持していない限り，例外処理が起動される。

b)　distance(other:TM_GeometricPrimitive):TM_Durationは，このTM_GeometricPrimitiveからもう一つのTM_GeometricPrimitiveまでの距離，つまり，それらの時間位置の差の絶対値を戻さなければならない。距離は二つのTM_GeometricPrimitiveの，最も近いTM_Position間の距離である。もし，いずれかのTM_GeometricPrimitiveが他方と連結，重複又は包含されている場合は，この操作は0を戻さなければならない。この操作は，もし（1）二つののいずれか一方が不確定な場合，（2）二つのTM_Positionが同じTM_ReferenceSystem（時間参照系）に関連しない場合，又は（3）いずれかのTM_PositionがTM_OrdinalReferenceSystemに関連する場合は，例外処理を起動する。

5.2.3.6　TM_Duration（時間長データ型）　TM_Duration（時間長データ型）（図5.5）は時間次元における長さ及び距離を記述するために使われるデータ型である。これは二つの下位型をもつ。

TM_PeriodDuration（期間長さデータ型）は期間の長さに関する情報を交換するためにJIS X0301及びISO 8601によって規定された形式を使用する。この形式によって期間の長さは，

年，月，日，時，分，秒の複数の時間単位用語を用いて具体的に表現できる。個々の単位に対する値は必須ではないが，少なくとも一つの単位に対する値は設定されなければならない。

属性:

a) Designator : CharacterString = P　以降の文字列が期間の長さを表現していることを示す必須の文字“P”

b) years ［0..1］ : CharacterString　期間の中の，年の数を表す正整数値と，その後に文字“Y”を加えた文字列

c) months ［0..1］ : CharacterString　期間の中の，月の数を表す正整数値と，その後に文字“M”を加えた文字列

d) days ［0..1］ : CharacterString　期間の中の，日の数を表す正整数値と，その後に文字“D”を加えた文字列

e) timeIndicator ［0..1］ : CharacterString =“T”　一連の指定に日より小さな単位の値が含まれる場合に，常にまず指定されなければならない文字“T”

f) hours ［0..1］ : CharacterString　期間の中の，時間の数を表す正整数値と，その後に文字“H”を加えた文字列

g) minutes ［0..1］ : CharacterString　期間の中の，分の数を表す正整数値と，その後に文字“M”を加えた文字列

h) seconds ［0..1］ : CharacterString　期間の中の，秒の数を表す正整数値と，その後に文字“S”を加えた文字列

最も右に指定された単位の値は，正整数値ではなく正の十進小数値で表現してもよい。

例　5日4時間30.7分という期間は，P5DT4H30.7Mと表わされる。

備考　この形式はJIS X0301及びISO 8601の中で，グレゴリオ暦及び協定世界時（UTC）の日付を使用するために定義されたものであるが，二つの時点が，年月日によって記述された暦と時分秒によって記述された時計を参照するときはいつも，長さ及び距離を表現するデータ型としてTM_PeriodDurationを用いてよい。

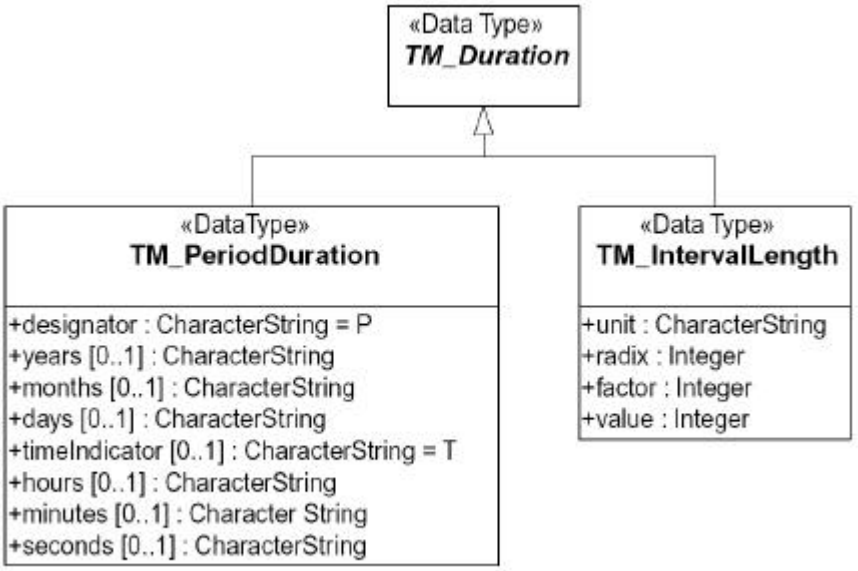

図5.5 － 時間長型

TM_IntervalLength（時間間隔データ型）は時間間隔について規定した ISO 11404 によって明示されたデータ型であり，この UML 図にあるように表現される。これは一つの時間単位の倍数を指定することによる時間長の表現を可能にする。

属性:

a) unit : CharacterString　時間間隔を表現するために使用された測定単位の名称を示す文字列

b) radix : Integer　時間単位となる乗数の基底となる正整数値

c) factor : Integer　基底のべき乗を行う指数を示す整数値

d) value: Integer　指定された単位値 radix $^{(-factor)}$ の倍数として時間間隔の長さを表す整数値

例　Unit = "second", radix = 10, factor = 3, value = 7 と指定することによって　7 ミリ秒（milliseconds）　という時間間隔を明示する。

5.2.4 時間位相オブジェクト

5.2.4.1 序論　位相はオブジェクト間の時間の連結性に関する情報を提供するものである。そして，このことから時間における地物の順序についての情報を提供することも可能である。他方，位相は，時間における位置に関する情報は提供しない。位相関係は幾何情報から導出できる場合もあるが，時間位置に関するデータは必ずしもこの導出には適していないため，位相について明確に記述することが必要となることもある。また，たとえそれが導出可能なものであっても，位相関係が明確に記述されていることが要件として必要となる際には，位相を用いることができる。

例　単一の順序年代において，幾つかの事象及び状態の順序を観測することは可能かもしれないが，順序時間参照系では，これらの事象及び状態について明確な時間への割り当ては行わない。これらの事象及び状態の順序は，位相プリミティブを用いてモデル化することによって記述できる。

5.2.4.2 TM_TopologicalPrimitive（時間位相要素）　TM_TopologicalPrimitive（時間位相要素）（図 5.6）は，TM_Node（時間ノード要素），TM_Edge（時間エッジ要素）の二つの下位型をもつ。応用システムが時間の連結性に関する情報とともに時間位置に関する情報をもつ場合，TM_TopologicalPrimitive は，同じ次元の TM_GeometricPrimitive（時間幾何要素）と関連をもつ。位相プリミティブは連結性に関する情報の提供を意図したものであるため，互いを結びつける関係こそがその最も重要な特性となる。また，すべての TM_TopologicalPrimitive は一つのそして唯一の TM_TopologicalComplex（時間位相複体）の要素でなければならない。

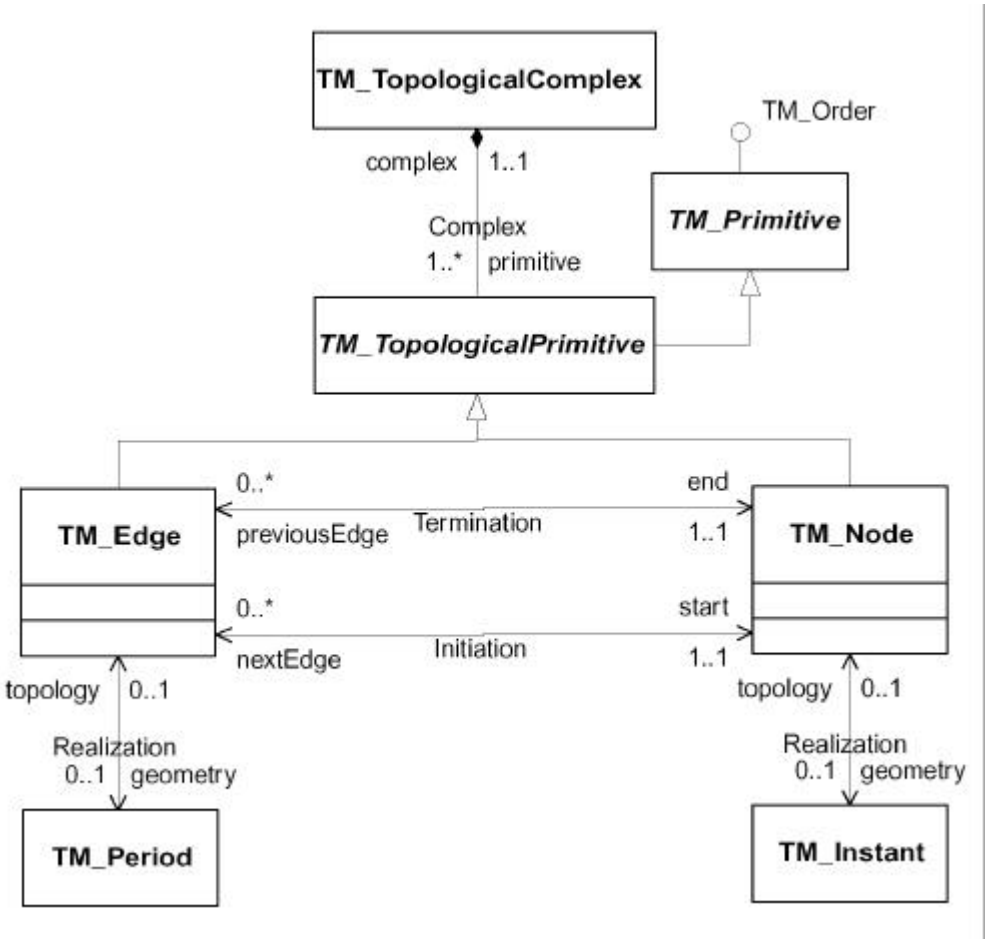

図 5.6 － 時間の位相

5.2.4.3　TM_Node（時間ノード要素）　TM_Node（時間ノード要素）は時間における0次元位相要素である。その幾何的実現として TM_Instant（瞬間要素）と対応する。

関連:　TM_Node は三つの関連を支持しなければならない。

a)　Initiation（開始）は，この TM_Node と，それを始点にもつ TM_Edge（時間エッジ要素）を結合しなければならない。

b)　Ternmination（終了）は，この TM_Node と，それを終点にもつ TM_Edge を結合しなければならない。

c)　Realization（実現）は，この TM_Node と，これに対応する TM_Instant とを結合する省略可能な関連である。一つの TM_Instant と関連することができるのは唯一の TM_Node であり，一つの TM_Node と関連することができるのは唯一の TM_Instant である。

5.2.4.4　TM_Edge（時間エッジ要素）　TM_Edge（時間エッジ要素）は時間における一次元位相要素である。TM_Edge は TM_Period（期間要素）と対応する。

関連:　TM_Edge は三つの関連を支持しなければならない。

a)　Initiation（開始）は，この TM_Edge と，その始点となる TM_Node（時間ノード要素）とを結合しなければならない。TM_Edge は一つのそして唯一の始点ノードを持たなければならない。

b)　Termination （終了）は，この TM_Edge と，その終点となる TM_Node とを結合しなけれ

ばならない。TM_Edge は一つのそして唯一の終点ノードを持たなければならない。

c) Realization（実現）は，この TM_Edge と，これに対応する TM_Period とを結合する省略可能な関連である。一つの TM_Period と関連することができるのは唯一の TM_Edge であり，一つの TM_Edge と関連することができるのは唯一の TM_Period である。

5.2.4.5　TM_TopologicalComplex（時間位相複体）　TM_TopologicalComplex（時間位相複体）は TM_Primitive（時間要素）の集合である。

関連:

a) Composition（構成）は，その TM_TopologicalComplex と，それに含まれる TM_TopologicalPrimitives（時間位相要素）の集合とを結合しなければならない。TM_TopologicalComplex 中の TM_Edge（時間エッジ要素）は二つの TM_Node（時間ノード要素）に結合するので，その複体中には最低二つの TM_Node が存在する。

5.2.4.6　線形グラフと非線形グラフ

5.2.4.6.1　非線形グラフ

TM_Edge（時間エッジ要素）の終端である Initiation 関連及び Termination 関連における多重度（図 5.6 参照）によって，非線形位相構造を表すことができる。ある TM_Node（時間ノード要素）は，複数の TM_Edge（時間エッジ要素）の開始ノード又は終了ノードになり得る。開始ノード又は終了ノードを共有する複数のエッジ要素は，異なった地物の時間特性，若しくは同じ地物における異なった時間特性を表現するために，何らかの方法で分離されたものとみなされる。

備考　時間の非線形位相は，空間における非平面位相に類似している。双方共，交叉又は重複しているように見える二つの位相要素は，更に高次の次元においては実際に分離したものであると考えられる。

5.2.4.6.2　線形グラフ　時間は一次元であるため，時間位相は線形グラフとして表現されるべきものである。線形位相において時間位相複体は，TM_Edge（時間エッジ要素）の次には TM_Node（時間ノード要素）が現れるような TM_Primitive（時間要素）の列となる。その列の最初の要素は，最初の TM_Edge の開始ノードとしての TM_Node であり，最後の要素は，最後の TM_Edge の終了ノードとしての TM_Node である。応用スキーマにおいて線形位相であるという制約を設けるためには，TM_Edge の Initiation 関連及び Termination 関連における終端の多重度を 0..1 に制限しなければならない。それによって，両端以外のすべての TM_Node は，必ず前のエッジと後のエッジという二つの TM_Edge を持たなければならないこととなる。

5.2.4.7　TM_Order（時間順序インタフェース）　TM_TopologicalPrimitive（時間位相要素）は TM_Primitive（時間要素）を通じて，TM_Order（時間順序インタフェース）を継承する。TM_Order は，この TM_Primitive の，他の TM_Primitive からの相対的な位置を測定するための操作を提供する。

RelativePosition (other:TM_Primitive) : TM_RelativePosition

二つの TM_TopologicalPrimitive の相対位置は，これらを含む TM_TopologicalComplex（時間位相複体）の TM_TopologicalPrimitive の列の中におけるそれぞれの位置に依存する。TM_TopologicalPrimitive に対するこの操作は，次に示す列挙型の TM_RelativePosition（時間相対位置データ型）（図 5.4）の値を返さなければならない。

戻り値	状況
Before	この TM_TopologicalPrimitive は，other で指定されたものより以前に順序づけられ，Initiation 関連及び Termination 関連で結びつけられていない。
Meets	二つの TM_TopologicalPrimitive は共通の TM_Node（時間ノード要素）と関連をもった TM_Edge（時間エッジ要素）であり，この TM_Edge は TM_Node の previousEdge として Termination 関連をもち，other で指定された TM_Edge は TM_Node の nextEdge として Initiation 関連をもつ。
Begins	この TM_TopologicalPrimitive は TM_Node，other で指定されたものは TM_Edge であり，二つは Initiation 関連で結びつけられている。
BegunBy	この TM_TopologicalPrimitive は TM_Edge，other で指定されたものは TM_Node であり，二つは Initiation 関連で結びつけられている。
Equals	この TM_TopologicalPrimitive と other で指定されたものは同一である。
Ends	この TM_TopologicalPrimitive は TM_Node，other で指定されたものは TM_Edge であり，二つは Termination 関連で結びつけられている。
EndedBy	この TM_TopologicalPrimitive は TM_Edge，other で指定されたものは TM_Node であり，二つは Termination 関連で結びつけられている。
MetBy	二つの TM_TopologicalPrimitive は共通の TM_Node と関連をもった TM_Edge であり，この TM_Edge は TM_Node の nextEdge として Initiation 関連をもち，other で指定された TM_Edge は TM_Node の previousEdge として Termination 関連をもつ。
After	この TM_TopologicalPrimitive は，other で指定されたものより以後に順序づけられ，Initiation 関連及び Termination 関連で結びつけられていない。

二つの TM_TopologicalPrimitive が同じ TM_TopologicalComplex のなかに存在しないとき，この操作は例外処理を起動する。

5.3　時間参照系

5.3.1　時間参照系の型　時間領域における値は，時間参照系との関係のもとに計測された時間上の点である。JISX0301 及び ISO 8601 は，情報交換のためにグレゴリオ暦及び 24 時間の地方時又は協定世界時（UTC）の使用を規定しており，地理情報においても，これを基本時間参照系として使用しなければならない。異なる時間参照系が地理情報の応用システムにとって適当と考えられる場合がある。その場合，地物カタログ又は，応用スキーマ及びデータ集合と関連するメタデータに，時間参照系について記述された文書の引用又はその参照系に関する記述が含まれていなければならない。単一の地物カタログ，応用スキーマ又はデータ集合の中で複数の時間参照系が使用されるときは，それぞれの時間特性の定義が，使用する時間参照系を明確に記述しなければならない。この箇条は，これらの記述のための基礎となる概念スキーマを記述する。附属書 C には，これらの記述の際に用いなければならないこのスキーマによって導かれたメタデータ要素を定義している。

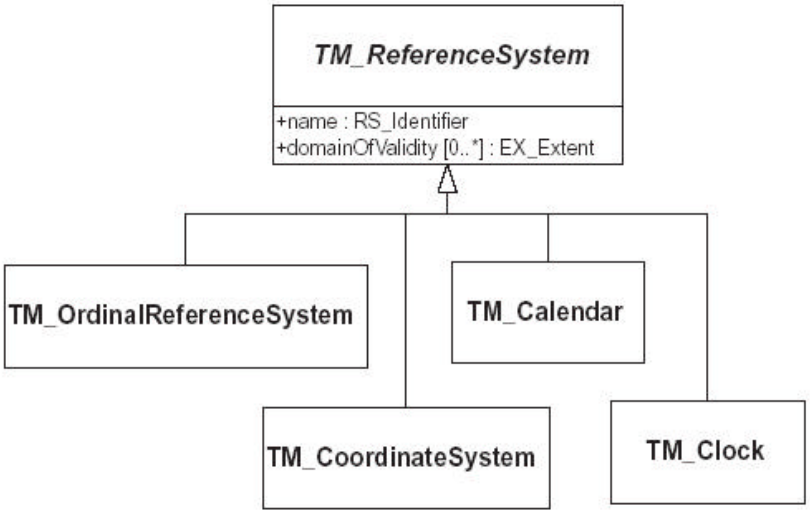

図 5.7 － 時間参照系

時間参照系パッケージは，時間参照系の三つの共通型，すなわち，暦（より高精度を求める場合は時刻と共に用いられる），時間座標系，順序時間参照系を含んでいる（図 5.7）。

TM_ReferenceSystem（時間参照系要素）は，以下の属性を含まなければならない。

a) name:RS_Identifier（名称：参照系識別子データ型）はこの時間参照系を他と識別するための名称を保持しなければならない。データ型 RS_Identifier は地理情報標準－座標値による空間参照で定義される。

b) DomainOfValidity:EX_Extent（有効範囲：範囲データ型）は，TM_ReferenceSystem が適用可能である空間と時間の範囲を明らかにしなければならない。データ型 EX_Extent は ISO 19103 において規定される。これは空間と時間の両方の範囲を記述することを許容する。有効範囲がデータ集合全体の範囲よりも小さいような TM_ReferenceSystem を参照する TM_Position（時間位置情報データ型）を応用スキーマが含む場合は，この属性が使用されなければならない。

以下の三つの箇条は，三つの参照系の型のスキーマを記述している。

5.3.2　暦と時法　暦と時法は共に間隔目盛の基礎である。暦は離散的な時間参照系であり，１日の精度で時間位置を定義するための基礎を提供する。時法は，一日のなかの時間位置を定義するための基礎を提供する。指定された日の中の時間位置を完全に記述するためには，暦と共に時法が用いられなければならない。図 5.8 は，TM_Calendar（暦要素）及び TM_Clock（時法要素）の詳細を示す。

暦はさまざまな複合的内部構造をもつものである。このスキーマは単純な外部の暦インタフェースを定義している。附属書 D には，暦の内部構造のより詳細な記述がある。

すべての暦は，年，月，日といった要素の集まりで暦日を構成するための，規則の集合を提供している。すべての暦において，年は暦年代の基点を定義する参照事象日から順に番号付けられた数字である。一つの暦は二つ以上の暦年代を参照してもよい。

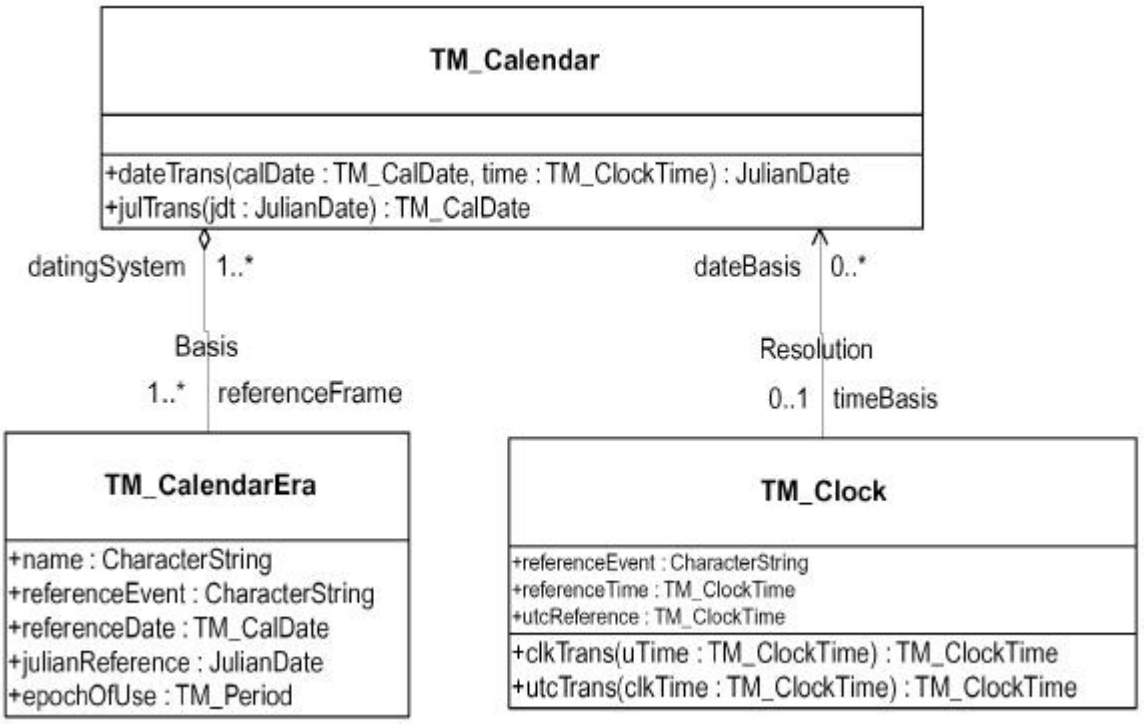

図 5.8 － 暦と時法

5.3.2.1　暦年代　TM_CalendarEra（暦年代要素）は，以下の属性を含まなければならない。

a)　name:CharacterString（名称：文字列）は，この暦に含まれる暦年代を他と識別する文字列でなければならない。

b)　referenceEvent :CharacterString（参照事象：文字列）は，暦年代の基点となる位置を決定する神話的又は歴史的事象の名称若しくは記述を，文字列で保持しなければならない。

c)　referenceDate :TM_CalDate（参照日付：カレンダー日付型）は，与えられた暦において参照事象の日を表した日付を保持しなければならない。多くの暦では，この日付は目盛の原点（すなわち初日）であるが，そうでない場合もある。

d)　julianReference :JulianDate（参照ユリウス日：ユリウス日付型）は，参照日付に対応するユリウス日付を保持する。

e)　epochOfUse :TM_Period（暦年期間：期間要素）は，この暦年代が日付の基礎として使用された期間が定義されなければならない。TM_Period（期間要素）における始点と終点のデータ型は JulianDate（ユリウス日付型）（5.4.5.2）でなければならない。

関連:

a)　Basis 関連は，この TM_CalendarEra を，この TM_CalendarEra を日付のための参照として用いている TM_Calendar（暦要素）と結びつけなければならない。

備考　TM_Calendar は TM_CalendarEra を一つ以上参照している。これは，例えば“和暦”という暦要素の中に，“昭和”“平成”などの暦年代要素を含むことを意味している。

5.3.2.2　暦　TM_Calendar（暦要素）は，以下の操作を提供しなければならない。

a) dateTrans　(calDate:TM_CalDate,time:TM_ClockTime):JulianDate　暦日変換（dateTrans）は，入力として，指定された暦における日付と指定された時法における時刻を受け付け，出力としてユリウス日付を返さなければならない。

b) julTrans (jdt:JulianDate):TM_CalDate　ユリウス日変換（julTrans）は，入力としてユリウス日付を受け付け，出力としてこの暦における日付を返さなければならない。

備考　ユリウス日の体系はそれ以外に知られているいかなる暦よりも古い原点をもつ時間参照系である。ユリウス日付を介した変換によって，比較的単純な原理で，ある暦から別の暦に日付を変換することができる。

個々の暦の内部構造の記述は，使用者がこれらの操作を実装することが可能となるための十分な情報を含まなければならない。それは暦に関連付けられる各暦年代の記述を含まねばならず，暦の日付をユリウス日付に対応づけるために必要な情報を提供しなければならない。

関連:

a) Basis 関連は，この TM_Calendar を，日付を割り当てるために参照する複数の TM_CalendarEra（暦年代要素）に結びつける。

b) Resolution 関連は，この TM_Calendar を，最小の暦間隔内の時点を特定するために用いられる TM_Clock（時法要素）に結びつける。

5.3.2.3　時法　TM_Clock（時法要素）は，以下の属性を含まなければならない：

a) referenceEvent:CharacterString（参照事象：文字列）は，太陽の正中又は日の出といった，時法の基点となる位置を決定する事象の名称若しくは記述を保持しなければならない。

referenceTime:TM_ClockTime（参照時刻：時刻要素）は，与えられた時法において表現される参照事象の時刻を保持しなければならない。参照時刻は通常時計目盛の原点であるが，そうでない場合もある。

備考1.　例えば，太陽の正中を 12 時とするのは，時計目盛の原点としていない例である。このような場合を受け入れるために，記述を追記している。

utcReference:TM_ClockTime（協定世界時参照：時刻要素）は，参照時刻に対応する 24 時制地方時又は協定世界時を提供しなければならない。

備考2.　日本国内における 24 時制地方時は国内標準時である。

TM_Clock は，以下の操作を提供しなければならない：

b) clkTrans(uTime:TM_ClockTime):TM_ClockTime　時刻変換（clkTrans）は，24 時制地方時又は協定世界時で表現された時刻を入力とし，この時法での時刻を返さなければならない。

c) utcTrans(clkTime:TM_ClockTime):TM_ClockTime　協定世界時変換（utcTrans）は，この時法で表現された時刻を入力とし，24 時制地方時又は協定世界時を返さなければならない。

5.3.3　時間座標系　暦日と時刻によって時点を指定すると，時点同士の距離の計算及び時間演算の関数表現が複雑になる。時間座標系はこの種の応用システムを支援するために使用されることがある。時間座標系は単一の時間間隔によって定義される連続な間隔目盛を基礎としなければならない。

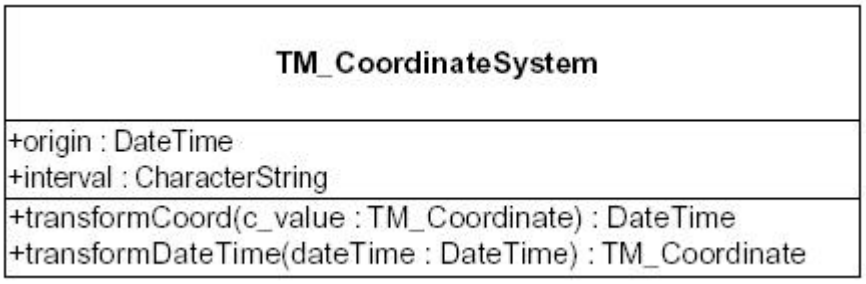

図 5.9 － 時間座標系

TM_CoordinateSystem（時間座標系）（図 5.9）は，二つの属性を含まなければならない：

a) origin:DateTime（原点：日時データ型）は，目盛の原点を保持しなければならない。原点はグレゴリオ暦による日付と協定世界時による時刻で規定されなければならない。DateTime は適当な精度に丸められる。

b) interval:String（間隔：文字列）は，目盛の基本的な間隔として使用される単一の計測単位の名前を保持しなければならない。時間間隔は応用システムにとって適切な間隔を選択してよいが，その指定は JIS Z 8203:2000 及び ISO 31-1 によって規定された時間の計測単位の中にあるもの，若しくは ISO 1000 で規定されたその組合せでなければならない。

TM_CoordinateSystem は，以下の操作を提供しなければならない：

c) transformCoord(c_value :TM_Coordinate) :DateTime 座標値変換（transformCoord）は，この時間座標系の座標値を入力とし，対応する日時をグレゴリオ暦及び協定世界時で返さなければならない。

d) transformDateTime(dateTime:DateTime):TM_Coordinate 日時変換(transformDateTime)は，グレゴリオ暦及び協定世界時で指定された日時を入力とし，この時間座標系の座標値を返さなければならない。

5.3.4　順序時間参照系　例えば地質学及び考古学などの多くの地理情報の適用事例において，その期間の長さよりも，時間における相対的な位置のほうが正確に知られている。順序時間参照系は，複数の事象について，それらの順序は確定できるが，その事象間の時間的間隔は正確に決定することができないような場合に適している。

順序時間参照系は順序性による目盛にその基礎を置いている。最も単純な形式では，ある順序時間参照系は順序付けられた事象の列である。一般的には，ある特定の事象の列が，一つの位置と関連づけられることになる。異なった位置の間の時間関係は，一つの位置で起きた複数の事象と別の位置における複数の事象との間の，時間的特性ではない相互関係の度合いによって決めることができる。このような相互関係によって，同様の事象が起こった期間というものについて定義するための，より広汎な基準に基づいた時間参照系を開発することができる。順序年代という言葉は，このような期間を参照するために，この標準で使用される用語である。

順序時間参照系は時間で順序付けられ，名前をもった時代の集合から構成される（図 5.10）。順序参照系は時として，ある与えられた階層レベルにおける順序年代がより詳細な連続した順序年代の列を含むというように，階層的に構造化される。

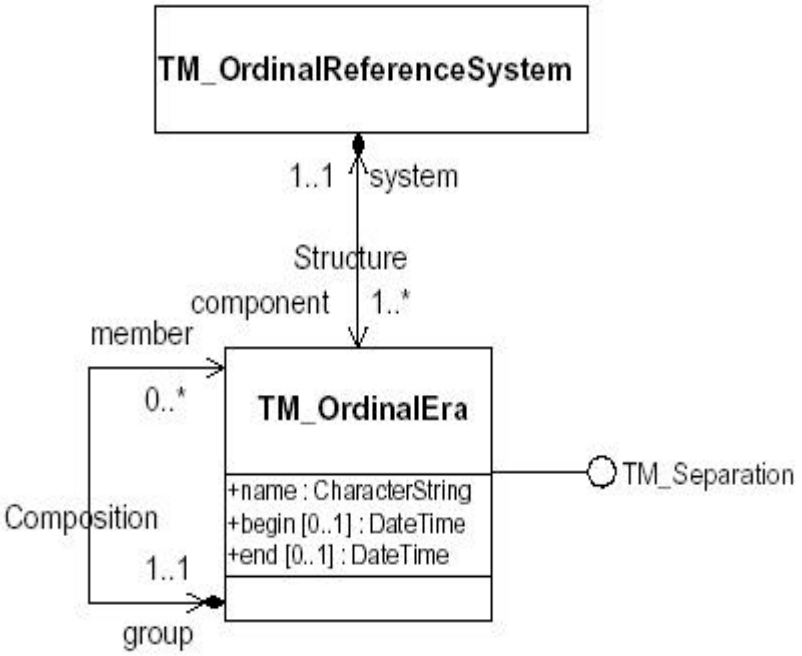

図 5.10 － 順序時間参照系

TM_OrdinalReferenceSystem（順序時間参照系要素）は TM_ReferenceSystem（時間参照系要素）から継承された属性だけをもつ。Structure（構造）関連は，階層の最上位レベルの TM_OrdinalEra（順序年代要素）の列を指している。

順序年代要素（TM_OrdinalEra）は，三つの属性を保有する：

a) name:CharacterString（名称：文字列）は，TM_OrdinalReferenceSystem の中において，この順序年代を他と識別する文字列でなければならない。

b) begin:DateTime（始点：日時データ型）は，順序年代が始まる時点が既知である場合に指定することができる。TM_Position（時間位置情報データ型）はグレゴリオ暦と協定世界時における時刻で示す DateTime（日時データ型）によって指定されなければならない。DateTime は JIS X0301 及び ISO 8601 で規定されているように，適当な精度レベルで丸めてもよい。

c) end:DateTime（終点：日時データ型）は，順序年代が終わる時点が既知である場合に指定することができる。TM_Position はグレゴリオ暦と協定世界時における時刻で示す DateTime によって指定されなければならない。DateTime は JIS X0301 及び ISO 8601 で規定されているように，適当な精度レベルで丸めてもよい。

TM_OrdinalEra は，TM_Separation（時間距離インタフェース）を支持してもよい（5.2.3.5 参照）。

ある TM_OrdinalEra は，Composition（構成）関連によって識別された，より短い TM_OrdinalEra の列の集まりであってもよい。

5.4　時間位置

5.4.1　序　時間における位置を特定する方法は，それぞれの時間参照系によって異なった型で規定される。以下の箇条では，それぞれの時間参照系において時点を特定するのに用いなければならないデータ型を定義する。

5.4.2　グレゴリオ暦と UTC で参照される時間位置　JIS X0301 及び ISO 8601 では，情報交換の目的におけるグレゴリオ暦と，24 時制地方時又は協定世界時の使用法について規定している。これらは地理情報において使用される基本的な時間参照系であらねばならない。地理情報標準－応用スキーマのための規則の附属書 E では，JIS X0301 及び ISO 8601 に準拠して日付と時刻を文字列化した形式により，日付及び日時に対する基本的なデータ型を定義している。論理的には，DateTime（日時データ型）は TM_Position（時間位置情報型）の下位型である。このデータ型はグレゴリオ暦と協定世界時（UTC）によって参照される時間位置を記述するために使用することができる。5.4.4 で定義しているデータ型は，日付と時刻として指定される数値を明記している。それらのデータ型は，グレゴリオ暦及び協定世界時（UTC）を含む任意の暦並びに時法で参照される時間位置として使用することができる。

5.4.3　TM_Position（時間位置情報データ型）　TM_Position（時間位置情報データ型）は四つの下位型（図 5.11）及び TM_ReferenceSystem（時間参照系要素）との関連をもつ抽象クラスである。すべての TM_Position は TM_ReferenceSystem と関連をもたなければならない。TM_Position は一つの属性をもつ。

属性:

indeterminatePosition: TM_IndeterminateValue（不確定位置：不確定時間位置表現値リスト）は，必須の属性値ではない。TM_Position の下位型がデータ型として使用されない場合，TM_Position はこの属性値だけを提供することになる。この属性が TM_Position の下位型において使用される場合，それは下位型によって提供された特定の時間位置情報に対する条件付けをおこなう。

列挙型データ型である TM_IndeterminateValue（不確定時間位置表現値リスト）は，不確定な位置に関する四つの値を提供する。

a)　“unknown”（未知）は親クラス TM_Position とともに使用され，時間位置の特定値が与えられていないことを示す。

b)　“now”（現在）は TM_Position の任意の下位型とともに使用され，既定値がアクセス時にはいつでも現時点に置き換えられなければならないことを示す。

c)　“before”（以前）は TM_Position の任意の下位型とともに使用され，実際の時間位置は分からないが，指定された値よりも前であることを示す。

d)　“after”（以後）は TM_Position の任意の下位型とともに使用され，実際の時間位置は分からないが，指定された値よりも後であることを示す。

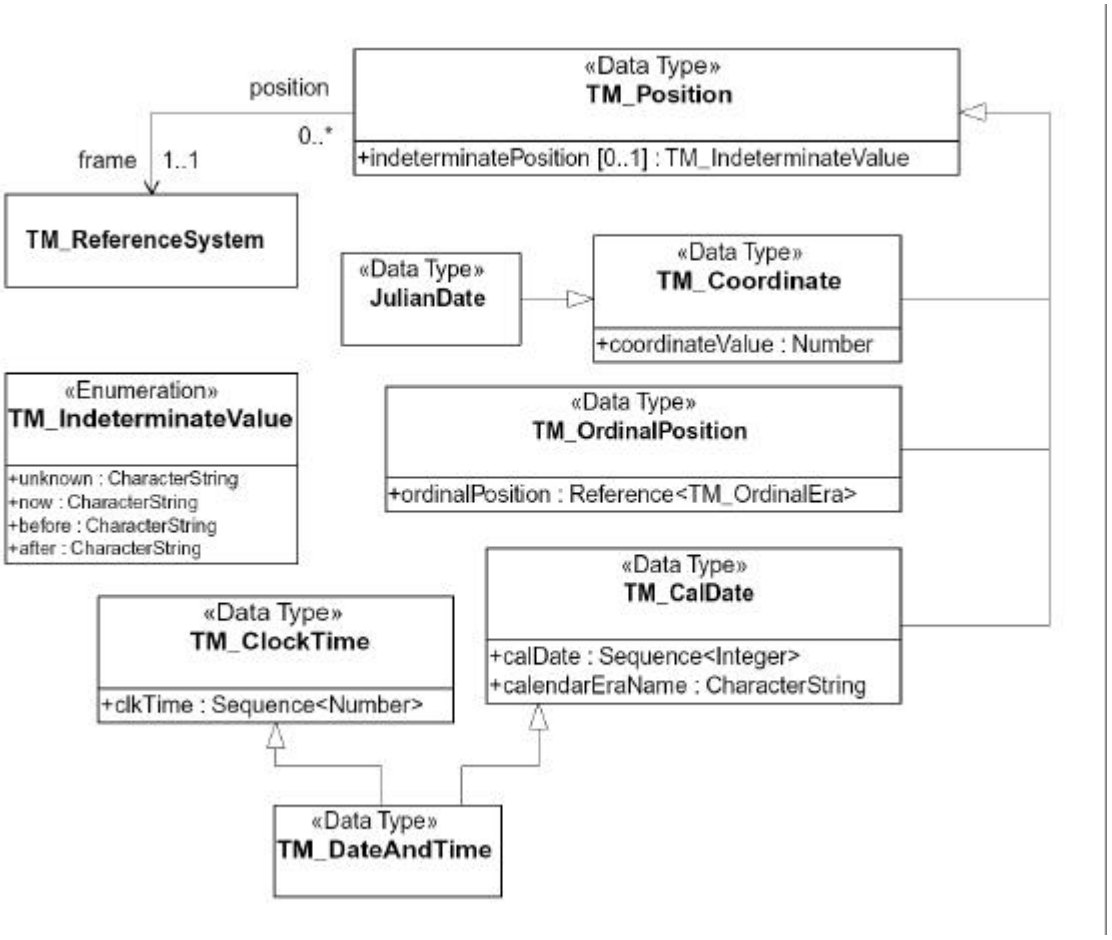

図 5.11 － 時間位置に関するデータ型

5.4.4　暦と時法で参照される時間位置

5.4.4.1　暦日　TM_CalDate（暦日データ型）は，ある暦の中で特定された時間位置を表すために使用されなければならないデータ型である。TM_CalDate は二つの属性をもつ：

a)　CalendarEraName ： CharacterString（暦年代名：文字列）は日付が参照する暦年代の名称を提供する。

b)　calDate ： Sequence <Integer>（日付：列<整数>）は正の整数の列で，最初の整数は暦における階層の最高位で使われる単位の，特定のインスタンスを示す。二番目の整数は暦における階層の二番目の階層で使われる単位の，特定のインスタンスを示す。以下同様に続く。JIS X0301 及び ISO 8601 で定義されているグレゴリオ暦における日付を示すための形式は，年，月，日の値によって構成されるあらゆる日付を示すために使用できる。

例　グレゴリオ暦において，整数列“1999,09,03”は 1999 年の第 9 の月の第 3 日として時間位置を示す。これは JIS X0301 及び ISO 8601 では“19990903”という形式で表わされる。

5.4.4.2　時刻　TM_ClockTime（時刻データ型）は，それだけでは時間位置を完全に示すことはできないので，TM_Position（時間位置情報データ型）の下位型ではない。この型は，一日よりも短い精度で時間位置を示すために，TM_CalDate（暦日データ型）と共に使用されなければならない。TM_ClockTime は一つの属性をもつ：

a)　clkTime : Sequence <Number>（時刻：列<数>）は日付（calDate）と類似した構造の，正の数の列で表わされる。最初の正数は，時法の階層の中で最高位に使われる単位の，特定のインスタンスを示し，二番目の数は二番目の階層に使われる単位となり，以下同様に続く。最後の数以外は整数なければならず，最後の数は整数又は実数でもよい。

例　現代の 24 時制では，列“22,15,30.5”は 22 番目の時の 15 番目の分から 30.5 秒たった時間位置を示す。これは JIS X0301 及び ISO 8601 では“221530.5”という形式で表わされる。

5.4.4.3　時刻を伴った日付　TM_DateAndTime（暦日時刻データ型）は，TM_CalDate（暦日データ型）と TM_ClockTime（時刻データ型）の両方のクラスの下位型であり，一日よりも短い精度で時間位置を識別する単一のデータ型を提供するために，両方の属性を継承する。

5.4.5　時間座標系で参照される時間位置

5.4.5.1　TM_Coordinate（時間座標データ型）　TM_Coordinate（時間座標データ型）は，ある時間座標系の中で特定された時間位置を表すために使用されなければならないデータ型である。TM_Coordinate は一つの属性をもつ：

a)　CoordinateValue:Number（座標値：数）は，時間座標系と関連して，原点からの標準間隔の乗数として表現される距離を保持する。

5.4.5.2　ユリウス日付　ユリウス日の体系は，ユリウス暦の紀元前 4713 年 1 月 1 日正午を原点にもつ時間参照系である。ユリウス日は整数値であるのに対し，ユリウス日付は小数値によってより細かい精度をもっている。

5.4.6　順序時間参照系で参照される時間位置　順序時間参照系では，瞬間の時間位置は，それが発生した順序年代なかで最短（最低階層）のものよりも短い長さであり，その中に含まれる。TM_OrdinalPosition（順序時間位置データ型）は，順序時間参照系の中で特定された時間位置を表すために使用されなければならないデータ型である。TM_OrdinalPosition は一つの属性をもつ：

a)　ordinalPosition: Reference <TM_OrdinalEra>（順序時間位置：参照<順序年代要素>）は，その瞬間が発生した順序年代への参照を提供する。

5.5　時間と地理情報の構成要素

5.5.1　地理情報構成要素の時間的な性質　**地理情報標準**－応用スキーマのための規則に記述された一般地物モデルは，地理情報のメタモデルを提供している。図 5.12 にあるように，そこには地物属性，地物操作，地物関係のメタクラスが定義されている。このそれぞれの構成要素が時間的な特性をもち得る。**地理情報標準**－応用スキーマのための規則では，ある応用スキーマにおいて，これらのメタクラスをインスタンス化するための規則を提供している。5.5.2 から 5.5.4 では，一般地物モデルにおける幾つかのメタクラスの時間的特性としての下位型を明示している。また，そこではこれらの構成要素が時間的情報をもつときに使用される規則の拡張も提供している。附属書 B には，これらのクラス及び規則が応用スキーマの中で使われるときの方法の実例を示している。

地理情報標準－地物カタログ化法は地物カタログを作成するための規則を規定しており，地物型，地物属性，地物操作，地物関係の定義を含んでいる。5.5.2 から 5.5.4 までは，これらの構成要素が時間的な性質をもつとき，これらを定義する際の必須事項を規定している。

地理情報標準－メタデータは地理情報と共に使用するメタデータ要素の標準的な集合を規定

しており，特定の応用システム及びプロファイルに対応して適切なメタデータ要素の追加定義を行うためのメカニズムを示している。5.5.5 は時間的な性質をもつメタデータ要素を定義するための必須事項を示している。

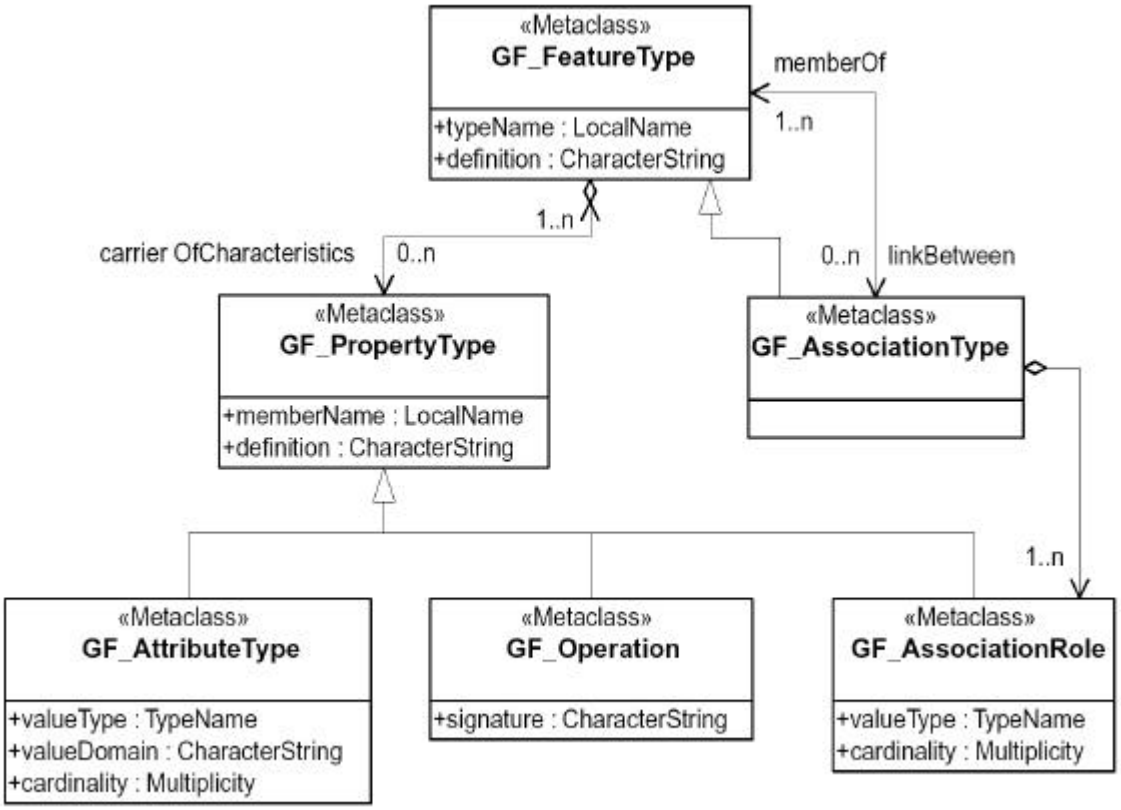

図 5.12 － 一般地物モデルのメタクラス

5.5.2　地物の時間属性　地物の時間属性は，時間位置に関連する静的な地物の特性を記述する。**地理情報標準**－応用スキーマのための規則で定義されている応用スキーマのための規則に準拠して，GF_TemporalAttributeType（時間属性型）は，GF_FeatureType（地物型）メタクラスをインスタンス化した地物自身のクラスの属性として，応用スキーマにおいて正しくインスタンス化されなければならない。**地理情報標準**－応用スキーマのための規則で規定された条件の下では，時間属性を属性とする地物型クラスに対応した UML クラスとしてインスタンス化されることになる。

静的な時間特性には，事象と状態の二種類がある（図 5.13）。

a)　事象はある瞬間に起こされた行為である。実際には，大部分の事象は短い時間間隔の中で発生するが，その間隔が計測のための解像度より相対的に短いときは，瞬間であるとする。事象についての GF_AttributeType.valueType（地物属性型の値型）は TM_Instant（瞬間要素），TM_Node（時間ノード要素）又は TM_Position（時間位置データ型）でなければならない。

b)　状態は，ある期間存続する地物又はデータ集合の特性であり，その状況である。その特性は地物属性及びメタデータ要素によって表現される。状態を記述する

GF_TemporalAttributeType（時間属性型）は二つの方法でインスタンス化できる。単純な場合は，地物型クラスの属性としてインスタンス化されなければならない。そのGF_AttributeType.valueType（地物属性型の値型）は TM_Period（期間要素）又はTM_Edge（時間エッジ要素）のいずれかでなければならない。より多くの情報が必要なときには，状態を表現するGF_TemporalAttributeType（時間属性型）は一つのUML クラスとしてインスタンス化されなければならない。そのクラスは， Beginning（始点）及びEnding（終点）を継承した TM_Period の下位型又は，Initiation（初期点）及びTermination（終端点）を継承したTM_Edge の下位型でなければならない。状態の特性については，そのクラスの一つ以上の属性によって記述されなければならない。その繰り返しは，地物型クラスとの関連における終端の多重度によって示されなければならない。時として，状態の変化は状態を開始又は終了する事象と関連する。その事象は，状態を表現するクラスの属性によって識別されなければならない。

事象は瞬間が複数回繰り返されたものとなることがある。また，状態は複数回繰り返されることがある。GF_AttributeType.cardinality（地物属性型の度数）では，応用スキーマにおいて許される繰り返し数を規定しなければならない。

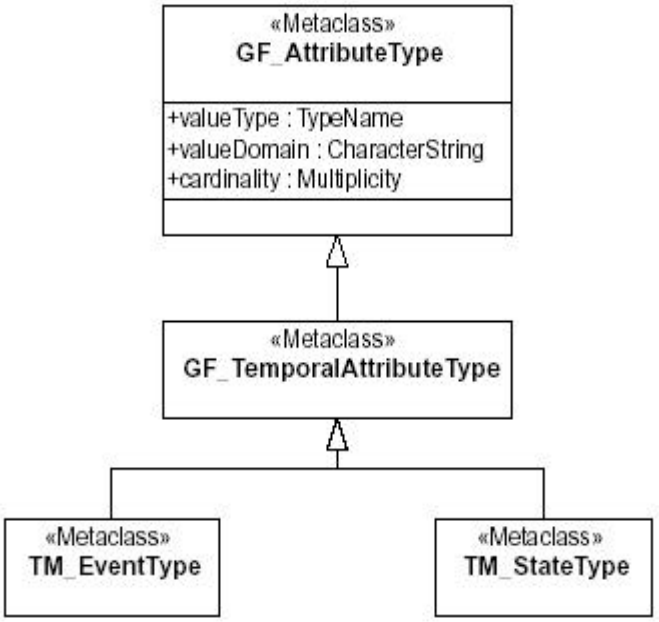

図 5.13 － 地物の時間属性型

事象及び状態は，しばしば規則的に繰り返される。連続する事象及び状態の間の間隔がその周期長である。地物の時間属性で繰り返し現象を記述するとき，応用スキーマの中では，地物型クラスに関連する UML クラスとしてインスタンス化されなければならない。このクラスは少なくとも二つの属性をもつ。その内の一つは，事象が起きた瞬間を示すか，属性の主題値が有効である期間の長さを示す。もう一つは，事象又は状態の発生の周期長を示す。周期長の値を表現するときは，TM_Duration（時間長データ型）（5.2.3.5 参照）を用いなければならない。

地理情報標準－地物カタログ化法は，地物カタログにおいて，地物属性の記述として含ま

なければならない要素について記述している。GF_TemporalAttributeType（時間属性型）が事象である場合，地物属性定義には，その時間位置で起こった動作とその結果が示されなければならない。GF_TemporalAttributeType が状態の場合は，地物属性定義には状態の特性が示されなければならない。地物の時間属性では，“Feature Attribute Value Domain Type”（属性値定義域型）の値は“not enumerated”（列挙型でない）となる。“Feature Attribute Value Domain”（属性値定義域），“Feature Attribute Value Data Type”（属性値データ型），“Feature Attribute Value Measurement Unit”（属性値計測単位）は，瞬間及び期間の時間位置を記述する上で基準とする時間参照系によって決定される。もし，時間参照系がグレゴリオ暦と UTC の組合せでない場合は，“Feature Attribute Value Domain”（属性値定義域）の記述では，使用される時間参照系を示さなければならない。“Feature Attribute Value Data Type”（属性値データ型）及び“Feature Attribute Value Measurement Unit”（属性値計測単位）はその時間参照系のもとで整合したものでなければならない。

5.5.3　時間地物操作　時間地物操作は動的なものであり，地物のもつ性質を表す値が時間の関数としての変化を表現する方法を記述するものである。時間地物操作は，地物操作のシグニチャ（FeatureOperation.signature）の中に時間を入力パラメタとして含んだ操作として，応用スキーマの中でインスタンス化されなければならない。

例　RoadIntersection（交差点）というクラスが与えられたとき，交差点を通る交通量は以下のような UML 操作として記述できる。

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

地理情報標準－地物カタログ化法は，地物カタログにおいて，地物操作を定義するための必要事項を規定している。時間地物操作の場合，Feature Operation Textual Description（地物操作のテキスト記述）には，操作の戻り値が時間の関数としてどのように変化するのか，時間の特性によってどのような影響を受けるのかについて記述されなければならない。また，時間が計測されるべき時間参照系も特定しなければならない。Feature Operation Formal Definition（地物操作の形式定義）では，独立変数の一つとして時間を含まなければならない。

5.5.4　時間と地物関係

5.5.4.1　地物関係の時間的な性質　地物関係は二つの方法で時間と関係する。ある地物関係は，互いの地物インスタンス間の時間的な関係にもとづいて存在する。また，別の地物関係は，関係として独自の時間特性をもつもので，さまざまな理由によって存在する。

例　同時に存在する二つの地物間の関係などが最初のケースであるといえる。このような時間関係は，他の特性をもたず，関係する地物の型に依存することもない。二番目のケースの例としては，ある期間，同じ所有者によって所有された二つの土地の間の関係がある。この場合，共通の所有者が関係における重要な要素であり，時間的な要素の重要性は低い。

時間地物関係は最初のケースに該当する。すなわち，それらの関係は，互いの地物インスタンス間の時間的な関係によって存在するのである。

5.5.4.2　時間地物関係

5.5.4.2.1　序　時間地物関係は，関連によって結びつけられた複数の地物の，それぞれの存続期間の間にある関係について明確に記述したものである。各地物が自己の存続期間を識別

する属性をもっている場合には，二つの地物の存続期間を比較する操作であるTM_Order.relativePosition（時間順序インタフェース：相対位置）を用いて時間地物関係を導出することは可能である。ある応用システムにとって時間地物関係が重要ならば，応用スキーマは規定する各地物に存続期間の属性を付与すべきである。時間位置が暦と時法又は時間参照系にもとづいて記述される場合，lifeSpan:TM_Period（存続期間属性：期間要素）を使用することができる。時間位置が順序時間参照系にもとづいて記述される場合は，lifeSpan:TM_Edge（存続期間属性：時間エッジ要素）を使用しなければならず，応用スキーマでは，存続期間を表現するすべての TM_Edge（時間エッジ要素）を含む非線型グラフとして，TM_TopologicalComplex（時間位相要素）をインスタンス化しなければならない。TM_Order（時間順序インタフェース）をサポートしない応用システムに対しても，地物時間関係を明示的に指定することができる。

この標準は時間地物関係の二つの下位型を規定する。それは単純時間地物関係と，地物継起である（図 5.14）。

5.5.4.2.2　単純時間地物関係　単純時間地物関係は二つ以上の地物間の相対的な時間位置を識別し，それ以上の意味はもたない。一般的には，この関係はすべての地物インスタンス間に存在するものであり，ある個別の地物型のためにあるものではない。

地理情報標準－応用スキーマのための規則で定義された応用スキーマのための規則によれば，地物関係は，地物型を表現する UML クラス間の関連としてインスタンス化されなければならない。しかしながら，純粋に時間的な地物関係は，通常地物型から独立している。もし応用システムが，特定の型をもつ地物の間の時間地物関係だけを考慮しているときには，地物型・クラス間の関連として表される時間地物関係のインスタンス化は適切であるといえる。しかし，すべての地物型のインスタンス間において，ある型の時間関係に関する情報をもつ必要のある応用スキーマにとっては，不十分な方式である。また，ある応用スキーマにとっては，5.5.4.2.1 で記述された各関連を導き出したほうがよい。それができない場合の代替手段として，スキーマに含まれるすべての地物型の上位型となる地物クラスのインスタンス化を考え，そのレベルにおける自分自身への関連として定義した地物関係のインスタンス化とすることが考えられる。

5.5.4.2.3　地物継起　地物継起とは，地物の集合が別の集合に置換されることである。置換とは，第１の集合の存続期間が終了した時点が，第２の集合の存続期間の開始時点となることを意味している。地物継起には空間及び時間の両方の局面があり，関連ある地物が，同じ空間的な位置を異なる時間に特定の順序で占めるということを表わしている。

地物継起は常に型に依存するというわけではない。すなわち，ある地物インスタンスの型が，常に決まった地物型のインスタンスで置換されるとは限らないということである。地物継起は，はん用地物レベルでモデル化できるが，地物型のレベルでモデル化できるとは限らない。

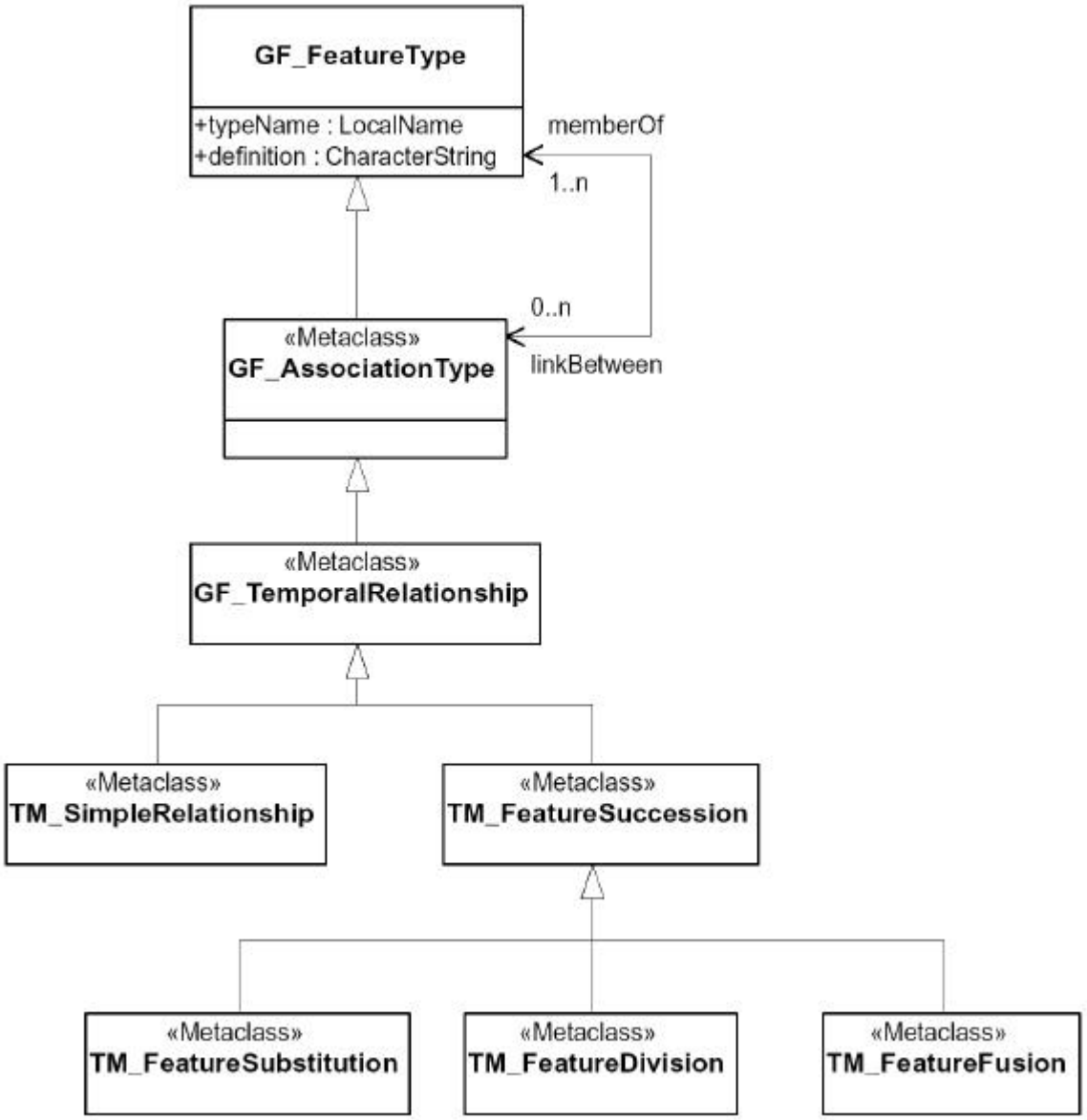

図 5.14 － 時間地物関係の分類

地物継起（TM_FeatureSuccession）には，地物置換（TM_FeatureSubstitution），地物分裂（TM_FeatureDivision），地物融合（TM_FeatureFusion）の三種類がある。地物置換は一つの地物インスタンスが，同じ又は異なる地物型の地物インスタンスに置換されることである。これは，二つの地物インスタンス間の一対一の関係を作る。地物分裂は一つの地物インスタンスが二つ以上の，同じ型の地物インスタンスへの分割が発生することである。これは，地物インスタンス間の一対多の関係を作る。地物融合は二つ以上の地物インスタンスが一つの地物インスタンスに併合されることである。これは地物インスタンス間に多対一の関係を作る。単一の事象によって，これらが組み合わされた継起の形式が生じることがある。その際，四つの組合せの可能性がある。

・分裂と置換

・融合と分裂
・融合，置換と分裂
・置換と融合

例　林の一部を空き地にして，これを駐車場にすることは，“分裂及び置換”の例である。牧草地（pasture）に隣接している林を開墾して，これらを合わせて牧草地を広げることは，“置換及び融合”の例である。

備考　一つの地物の特性が変化することは，それ自体では地物継起とみなすことができない。例えば，賃貸者数属性（numberOfOccupants）をもつ建物という地物型を考える。この属性の値はたえず更新されるが，それによって，ある建物インスタンスが別の建物インスタンスに置換したとは考えない。ある地物インスタンスが以前から存在した地物インスタンスに置き換わったとみなされるために必要となる変化の度合いは，応用システムに依存する。一般的な規則としては，地物の識別子が変わったときに置換が起こると考えられる。

継起関係は，TM_Order.relativePosition（相対位置操作）を，**地理情報標準**－空間スキーマで定義された一つ以上の空間操作とともに使用することによって導き出すことができる。

地物継起型における時間地物関係は，地物クラス間の UML 関連として又ははん用地物クラスの自己関連として，応用スキーマの中でインスタンス化される。名前，役割及び多重度はそれぞれの継起の型によって異なったものでなければならない。役割名は一つの地物が別のものに引き継ぐ順序を示さなければならない。継起が起きた時間を含めるためには，応用スキーマは発生時刻を示す属性のついた UML 関連クラスとして継起関係を表現しなければならない。

5.5.4.3　地物関係の時間特性　地物関係の時間特性は通常，時間関係の開始時間（began）又は終了時間（ended）によって瞬間に関する情報を与えるか，時間関係の存続期間に関する情報を与える。時間地物属性及び時間メタデータ要素と同じように，これらは事象又は状態を記述する。時間特性をもった地物関係は，UML 関連クラスとしてインスタンス化されなければならない。時間特性は，TM_Event（事象要素）又は TM_State（状態要素）メタクラス（5.5.2 参照）をインスタンス化する属性として表現されなければならない。

5.5.5　時間メタデータ要素　**地理情報標準**－メタデータは地理情報のための標準的なメタデータ要素の集合を定義している。また，応用スキーマ（図 5.15）の中で補助的なメタデータ要素を定義するための方法も規定している。時間メタデータ要素は時間地物属性と類似している。両者とも，時間位置に関連する静的な特性，すなわち事象又は状態を記述するが，地物属性が地物インスタンスの抽象として実世界の特性を記述するのに対し，メタデータ要素はデータの特性を記述する点が異なる。メタデータ要素によって記述されるデータの適用範囲は，複数のデータ集合の集まりに関するものから，一つの地物の，一つのの特性に関するものまでの幅をもつことができる。

時間メタデータ要素が事象を記述するとき，名前及び定義にはその時間位置を規定する動作と精度を示さなければならない。データ型属性（dataType）の型名（TypeName）は TM_Instant（瞬間要素）又は TM_Position（時間位置データ型）のいずれかでなければならない。

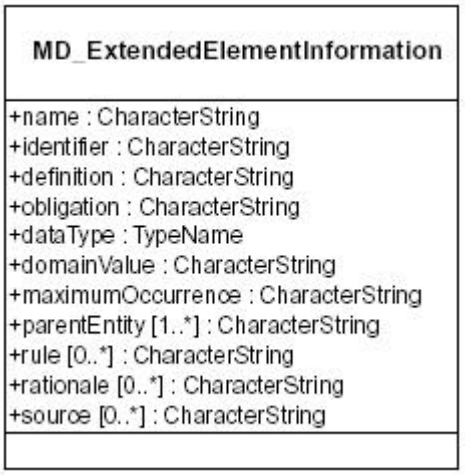

図 5.15 － 拡張されたメタデータ要素

時間メタデータ要素が状態を記述するとき，名前及び定義は状態の特性を記述しなければならない。データ型属性（dataType）の型名（TypeName）は TM_Period（期間要素）でなければならない。もし時間参照系がグレゴリオ暦と UTC の組合せでないときは，その定義は，使用する時間参照系の識別が可能となるようにしなければならない。

附属書 A（規定）　抽象試験項目群

A.1　応用スキーマ

a)　**試験目的**　応用スキーマには地物の時間特性が定義されており，メタデータ要素は規定された必要条件に従っていることを確かめる。

b)　**試験方法**　応用スキーマ又はプロファイルの文書を検査する。

c)　**参照**　5.

d)　**試験種類**　機能

A.1.1　データ交換のための応用スキーマ

a)　**試験目的**　応用スキーマには時間属性が定義されており，地物の時間関係及びメタデータ要素は規定された必要条件に従っていることを確かめる。

b)　**試験方法**　応用スキーマ中に含まれる時間属性と地物の時間関係の表現に関する定義が，時間属性又はその値を表現する時間オブジェクトの使用に対する必要条件を満たしたものであることを検査すること。応用スキーマの中で定義されている，どの時間メタデータ要素も必要条件を満たしていることを保証すること。すべての必要な属性及び時間オブジェクト間の関連が実装され，オプションの属性又は関連が必要条件に従って実装されていることを保証すること。必要なデータ型が時間位置の値のために使用されていることを確かめる。

c)　**参照**　5.2，5.4，5.5.2，5.5.4.2，5.5.5

d)　**試験種類**　基本

A.1.2　操作を伴うデータに関する応用スキーマ

a)　**試験目的**　応用スキーマは，時間地物操作を定義する A.1.1 の要件を満たすものであり，それが指定された必要条件に従って時間地物の操作が定義され，また，時間オブジェクトの操作が実装されていることを確かめる。

b)　**試験方法**　応用スキーマに含まれる時間地物操作の表現が，指定された必要条件に従っていることを保証するものであることを検査する。応用スキーマに含まれる地物の時間属性の表現が，時間属性又はそれらの値を表現するために使用される時間オブジェクトのいかなる操作も，必要条件に従って実装されていることを検査する。

c)　**参照**　A.1.1，5.2，5.5.3，5.5

d)　**試験種類**　基本

A.2　地物カタログ

a)　**試験目的**　地物カタログは規定された必要条件に従って時間特性が定義されていることを確かめる。

b)　**試験方法**　カタログを調べる。時間地物属性，時間地物操作又は時間地物関係の定義が必要条件に従ったものであることを点検する。時間参照系に関する識別と記述が満足なものであることを確かめる。

c)　**参照**　5.3，5.5.2，5.5.3，5.5.4.2

d)　**試験種類**　基本

A.3　メタデータ要素の仕様

a)　**試験目的**　時間メタデータ要素が規定された必要条件に従って定義されていることを確かめる。

b)　**試験方法**　メタデータの仕様におけるメタデータ要素定義が，イベントや状態にとって適切な必要条件に従ったものであることを検査する。

c)　**参照**　5.5.5

d)　**試験種類**　基本

A.4　データ集合のためのメタデータ

a)　**試験目的**　データ集合のためのメタデータには，時間参照系に関して必要とされる情報が規定されていることを確かめる。

b)　**試験方法**　メタデータを検査する。もしメタデータが，それが記述するデータの中に時間位置がグレゴリオ暦及び協定世界時を参照しないものがあることを示している場合は，そのデータで使われている時間参照系の引用又は記述のいずれかが提供されていることを点検する。各記述において，すべての必要要素が含まれていることを確かめる。

c)　**参照**　5.5.5

d)　**試験種類**　基本

附属書 B（参考）　応用スキーマにおける時間の使用

B.1　時間地物属性

B.1.1　データ型としての TM_GeometricPrimitive（時間幾何プリミティブ）　図 B.1 は，応用スキーマにおける TM_GeometricPrimitive（時間幾何プリミティブ）の使用法を示している。地物型 Building（建物）は UML クラスとして表現されている。このクラスは，TM_Instant（瞬間要素）型の値をとる属性 dateOfConstruction（建設日）と，TM_Period（期間要素）型の値をとる属性 periodOfOccupancy（入居期間）をもっている。このスキーマでは，各 TM_GeometricPrimitives は，インタフェース TM_Order 及び TM_Separation で定義された操作を実装してはいない。TM_Instant は唯一の属性 TM_Instant.position : TM_Position をもつので，そのデータ型 TM_Position，又はその下位型の内の一つ，例えば DateTime のようなデータ型が，Building.dateOfConstruction のためのデータ型として使用される。

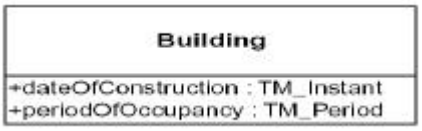

図 B.1 － データ型としての TM_GeometricPrimitive

B.1.2　時間属性としての TM_GeometricPrimitive　図 B.2 は，時間地物属性としての TM_GeometricPrimitive の，別の使用法を示している。この場合，地物属性 periodOfOccupancy は UML クラスとして表現され，UML の関連によって Building と結合している。PeriodOfOccupancy は TM_Period の下位型である。これは TM_Period から属性 begin 及び end を継承するが，両者はそれぞれデータ型 DateTime をとるように制限している。なお，TM_Period の定義によれば，これらの属性のデータ型は TM_Position となる。また，インタフェース TM_Order を継承した操作 relativePosition(other : TM_Primitive):TM_RelativePosition を使用し，インタフェース TM_Separation を継承した Length():TM_Duration 及び distance(other:TM_GeometricPrimitive):TM_Duration を使用している。

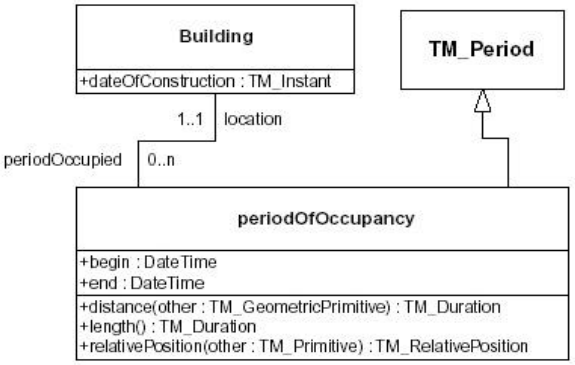

図 B.2 － 時間地物属性としての TM_GeometricPrimitives

B.1.3　属性としての TM_TopologicalComplex（時間位相複体）　図 B.3 は時間地物属性としての TM_TopologicalComplex の使用法を示している。考古学上の遺跡の歴史は，その遺跡が，ある文化圏の住人によって使用されたり又は使用されなかったりした期間の列として記述されることがしばしばある。これらの期間は，相互の順序は分かってはいるが，時間位置は分からない。このような歴史は TM_ToplogicalComplex として記述することができる。UML クラスとして表現された地物型 ArchaeologicalSite は地物属性 SiteHistory を表現するもう一つの UML クラスに関連する。SiteHistory は TM_TopologicalComplex の下位型である。またそれは，TM_Edge の下位型 OccupancyInterval の集約をもっている。

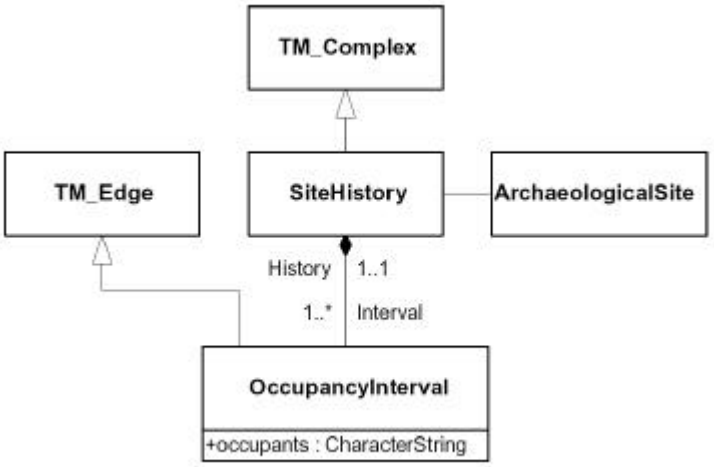

図 B.3 － 時間地物属性としての TM_TopologicalComplex

B.1.4　循環性のある属性値　図 B.4 は，周期的に循環する値をもつ地物属性の例を示している。地物型 Field（畑）は，0 から n の多重度をもつ UML クラスとして表現された属性 CropRotation（輪作作物）をもつ。この畑に植えられる作物（crop）の種類ごとに，このデータ型の一つのインスタンスが存在する。CropRotation は三つの属性をもつ。

・cropType は，作物の種類を示す
・frstPlanting は，この畑に初めて植えられた日付を示す
・rotationInterval は，この作物の栽培周期長を示す

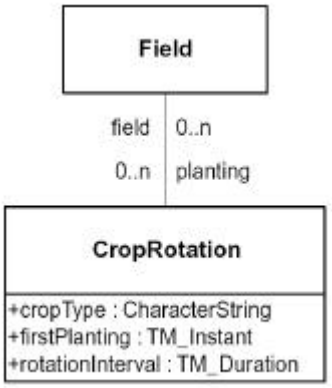

図 B.4 － 周期性のある属性値

B.2　時間地物関係

B.2.1　単純な時間関係　図 B.5 は，UML の関連によって実装された単純な地物関係の例であ

る。このスキーマは，サービスステーションの前には道路があることを示している。

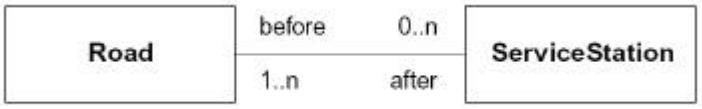

図 B.5 － 単純な時間関係の明示的表現

図 B.6 に示された Building の単純なスキーマは，操作 TM_Order.relativePosition を使用することによって単純な地物関係の導出が必要とされるデータの例である。

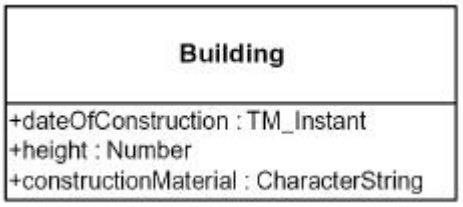

図 B.6 － Building スキーマ

このスキーマによって作成されたデータ集合に次の表 B.1 に示すような属性値をもった Building の四つのインスタンスがあるものとする。

表 B.1 － Building スキーマのデータ集合

Building 建物	constructionMaterial 建築材料	height 高さ	dateOfConstruction: TM_Instant.position [as Date] 建設日
A	brick（煉瓦）	45 m	1982
B	steel（鉄骨）	4 m	1967
C	Wood（木造）	8 m	1941
D	brick（煉瓦）	30 m	1967

TM_Instant は，インタフェース TM_Order から継承した操作 relativePosition (other:TM_Primitive) : TM_RelativePosition を使用することができる。もしこの操作が，Building.dateOfConstruction : TM_Instant の各インスタンスのもとで，他の各地物インスタンスを引数（つまり入力パラメタ other の値）として実行されると，それぞれ表 B.2 に示す値を返す。

表 B.2 － Building インスタンス間の単純な時間関係

	Target Instance			
Source Instance	A	B	C	D
A	Equal	After	After	After
B	Before	Equal	After	Equal
C	Before	Before	Equal	Before
D	Before	Equal	After	Equal

B.2.2　地物継起　図 B.7 は，地物型クラス間の関連としてモデル化された地物継起の例である。これは生態学的な継起の型の例であり，古い畑地の継起としてアメリカ東部では良く知られているものである。もし他から干渉されないままであれば，ある場所ではこの継起循環が生じる。これは地物型レベルでの地物継起をモデル化する際の問題点を示している。いつであっても，火事，嵐又は人間の介在が入ると，この継起は中断されて，以前の段階に戻ったり，既存の地物型を思いがけない地物型に変えてしまったりすることになる。

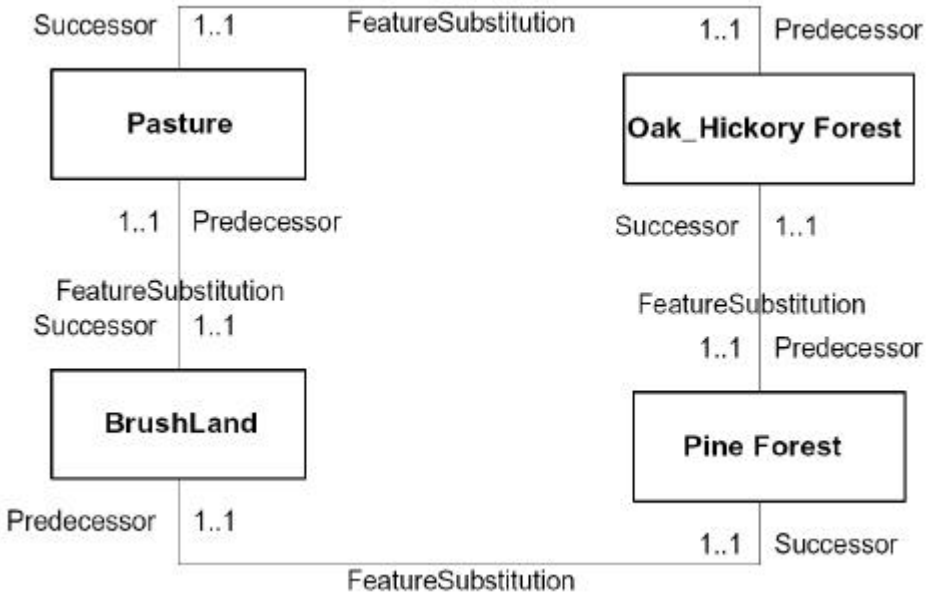

図 B.7 － 地物型間での地物継起

図 B.8 は，継起関係をもつさまざまな地物型の上位型が UML クラスの関連によって自己参照する地物継起モデルの例である。このようなモデル化は，どの地物型のインスタンスがどのインスタンスに継起するか予測できないときに必要となる。この例では，それぞれの関連は，継起が起きた時間を関連の属性によって記述することができるように，関連クラスとして表現されている。

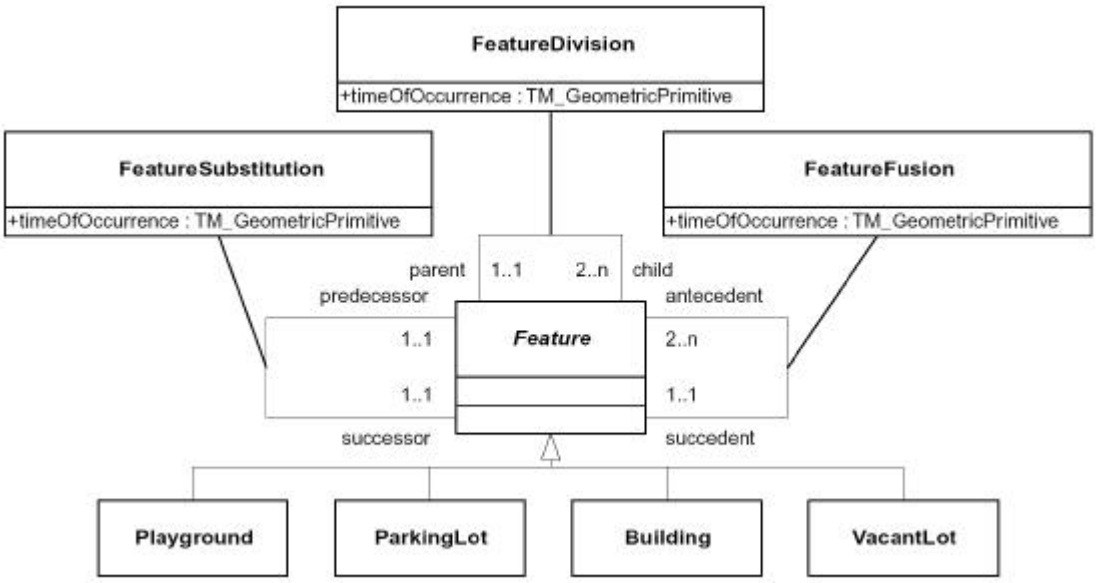

図 B.8 － 汎化した地物レベルでの地物継起

B.3　時間特性をもった地物関係　図 B.9 は牧場の本部と，それが管理し賃貸している放牧地（rangeland）との関係を示す応用スキーマである。RangelandTract（放牧区域）は Lease（賃貸）によって RanchHeadquarters（牧場本部）と関連する。賃貸は地物関係の例であり，時間関係とは考えられない。なぜなら，時間はこの関係において主要な性質ではないからである。しかしながら，賃貸は与えられた期間の間，効力をもつ。Lease はこのモデルでは UML 関連クラスとして表現され，関連に付与されるべき属性をもつことが許されている。この場合，それは，賃貸が効力をもつ期間を示す属性 leasePeriod をもつこととしている。そこにはまた，賃貸料などの他の属性ももつことができる。

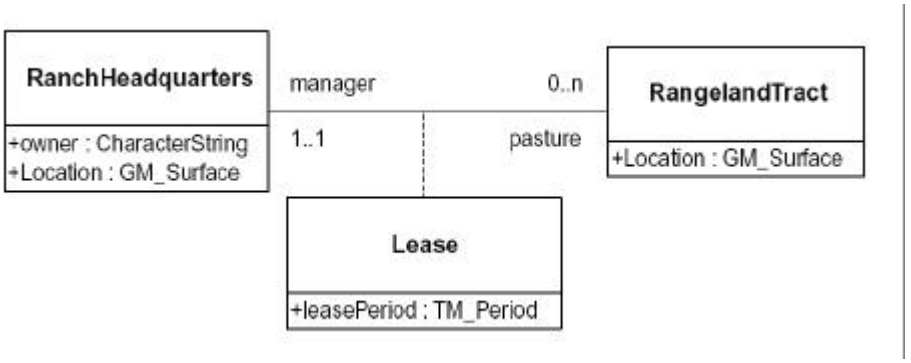

図 B.9 － 放牧地の賃貸

附属書 C（規定）　メタデータにおける時間参照系の記述について

C.1　時間参照系に関するメタデータ　本体の 5.3.1 は，グレゴリオ暦及び協定世界時以外の時間参照系を使用するデータ集合に関連するメタデータが，それらの時間参照系を記述する文書の引用を提供するか又はメタデータ内にそれらの時間参照系の記述を包含することを要求している。表 C.1 で規定されるメタデータ要素はこの目的のために使用される。この表の構造は地理情報標準－メタデータ，附属書 B に沿ったものである。

表 C.1 － 時間参照系を記述するためのメタデータ要素

	名称	定義	必須(M) / 条件(C) / 選択(O)	最大多重度	モデル要素やデータ型	領域
1	TM_ReferenceSystem 時間参照系要素	時間参照系に関する情報	C/ データ集合の時間情報がグレゴリオ暦を参照しない場合	N	class	Lines 2-33
2	name 名称	時間参照系を識別する名称	M	1	RS_Identifier	地理情報標準－座標による空間参照
3	DomainOfValidity 有効範囲	この時間参照系が使用される空間と時間の限界	C/ 時間参照系の範囲が，それを使用するデータ集合の範囲より小さい場合	N	EX_Extent	ISO 19103
4	Subtype 下位型	記述対象となる時間参照系の下位型	M	1	Specialization	TM_Calendar, TM_Clock, TM_CoordinateSystem, TM_OrdinalReferenceSystem
5	TM_Calendar 暦要素	Description of a calendar 暦の記述	C/ 下位型 TM_Calendar の場合	1	Class	Lines 6 – 15
6	dateTrans 暦日変換	この暦の日付をユリウス日付に変換する操作に関する記述	M	1	CharacterString	Free text

	名称	定義	必須(M) / 条件(C) / 選択(O)	最大多重度	モデル要素やデータ型	領域
7	julTrans ユリウス日変換	ユリウス日付をこの暦の日付に変換する操作に関する記述	M	1	CharacterString	Free text
8	Basis 基準暦年代	この暦と関連づけられた暦年代	M	N	Association	TM_CalendarEra
9	Resolution 暦分解	一暦日の中の時間位置を定めるために，この暦とともに使用する時法	O	1	Association	TM_Clock
10	TM_CalendarEra 暦年代要素	各暦年代の特性	M	N	Class	Lines 11-15
11	name 名称	この暦年代を識別する名称	M	1	CharacterString	Free text
12	referenceEvent 参照事象	この暦年代の基点となる事象	M	1	CharacterString	Free text
13	referenceDate 参照日付	この暦による参照事象の日付	M	1	TM_CalDate	Date in the calendar being described
14	julianReference 参照ユリウス日	参照事象のユリウス日付	M	1	JulianDate	Number
15	epochOfUse 暦年期間	この暦年代が日付の基礎として使用された期間	M	1	TM_Period	ISO 8601
16	TM_Clock 時法要素	時法についての記述	C/ 下位型 TM_Clock が指定されているか，暦分解が null でない場合	1	Class	Lines 17-21
17	referenceEvent 参照事象	この時法の基点となる事象	M	1	CharacterString	Free text

	名称	定義	必須(M) / 条件(C) / 選択(O)	最大多重度	モデル要素やデータ型	領域
18	ReferenceTime 参照時刻	この時法による参照事象の時刻	M	1	TM_ClockTime	Time in the clock being described
19	utcReference 協定世界時参照	UTC による参照事象の時刻	M	1	TM_ClockTime	ISO 8601
20	utcTrans 協定世界時変換	この時法の時刻を UTC 時刻に変換する操作に関する記述	M	1	CharacterString	Free text
21	clkTrans 時刻変換	UTC 時刻をこの時法の時刻に変換する操作に関する記述	M	1	CharacterString	Free text
22	TM_CoordinateSystem 時間座標系要素	時間座標系に関する記述	C/ 下位型 TM_CoordinateSystem の場合	1	Class	Lines 23-26
23	origin 原点	この時間参照系目盛の原点の位置をグレゴリオ暦と UTC 時刻で表現したもの	M	1	DateTime	ISO 8601
24	interval 間隔	この参照系の軸で時間の標準単位とされる間隔	M	1	CharacterString	ISO 31-1, ISO 1000
25	transformCoord 座標値変換	この時間座標系の座標値をグレゴリオ暦と UTC 時刻に変換する操作に関する記述	M	1	CharacterString	Free text
26	transformDateTime 日時変換	グレゴリオ暦と UTC 時刻で指定された日時をこの時間座	M	1	CharacterString	Free text

	名称	定義	必須(M) / 条件(C) / 選択(O)	最大多重度	モデル要素やデータ型	領域
		標系の座標値に変換する操作に関する記述				
27	TM_OrdinalReferenceSystem 順序時間参照系要素	順序時間参照系に関する記述	C/ 下位型 TM_OrdinalReferenceSystem の場合	1	Class	Lines 28-33
28	Structure 構造	この順序時間参照系の最上位レベルを構成する順序年代	M	1	Association	TM_OrdinalEra
29	TM_OrdinalEra 順序年代要素	Description of an ordinal era 順序年代に関する記述	M	N	Class	Lines 30-33
30	name 名称	この順序年代を識別する名称	M	1	CharacterString	Free text
31	begin 始点	この順序年代が始まる時点	O	1	DateTime	ISO 8601
32	end 終点	この順序年代が終わる時点	O	1	DateTime	ISO 8601
33	Composition 構成	この順序年代をさらに分割する順序年代	M	1	Association	TM_OrdinalEra

備考 UML 操作に関連するメタデータ要素は，テキストによる操作の記述である。

附属書 D（参考）　暦の記述

D.1　暦の内部構造　暦は離散的な時間参照系であり，一日の精度で時間位置を定義するための基礎を規定するものである。グレゴリオ暦は国際的な，事実上の標準規格であり，これを地理情報と共に使用できるよう組み込まれる必要がある。しかし，グレゴリオ暦とともに，いろいろな伝統的・歴史的な暦が存在する。それらは，いろいろな地理情報利用の上で適切なものである。例えば，考古学的な資料は，検討対象となる文化において使われていた暦によって，より正確に情報化できる。この標準の 5.3.1 は，グレゴリオ暦以外の暦を使用したすべてのデータ集合について，そのメタデータには，その暦についての記述又は記述の引用を含まなければならないということを要求している。この附属書では，いくつかの暦における記述の際に考慮する必要のある事項について述べている。

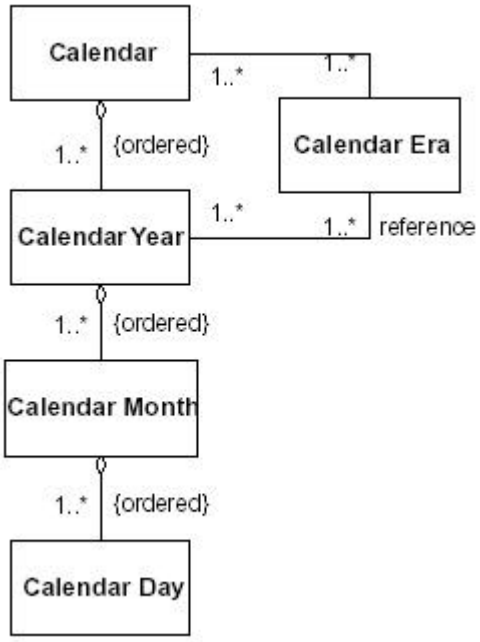

図 D.1 － 典型的な暦の内部構造

暦は階層的な構造をもち（図 D.1），規定された型の時間間隔が，それぞれのレベルで使用される。典型的な例では，階層中の一つのレベルにおける時間間隔の一連のインスタンスは，一回循環すると一段階上位の時間間隔となるような名称又は順序数がつけられる。ほとんどの暦は年と日を標準的な時間間隔として含んでいるが，中間レベルには月以外の時間間隔が使われることもある。ある暦では階層の中にさらに別のレベルをもつことがあり，また，他のレベルと並存したレベルをもつこともある。暦において，ある日とは，各階層のそれぞれのレベルで一つの時間間隔のインスタンスを識別したものである。

例　グレゴリオ暦の日付は，ある年の，その年の中のある月の，そして，その月の中のある日として識別されたものである。

暦は，複数の天体の周期を基準として各時間間隔が定められるが，その周期は互いに整数比とはならないため，複雑なものとなる。暦の循環周期と天体の循環周期との同期を維持するために，より低いレベルに“うるう”を挿入することによって間隔の長さを調整する。

a)　すべての暦の基本的な時間間隔は日であり，これは通常，地球が地軸を一回りする周期にほぼ一致する。暦で使用されるその他の時間間隔は日の整数倍である。多くの暦では，個々の日付は，月の中の順序数によって示される。ある種の暦ではそれ以外の方法が使われる。

b)　暦年は，太陽の周りを回転する地球の循環周期にほぼ一致する。この周期の長さは一日の整数倍ではないため，多くの暦では，暦年の長さの長期にわたる平均が地球の回転の周期と等しくなるように，一暦年の長さが変動することを許す設計となっている。そして暦は，それに応じた精度の中で変動する。多くの暦では，補助的な月又は日が，基本的な周期の中に挟み込まれる。多くの古代の暦では，“うるう”の挿入は，政治的又は宗教的な権威がその必要性を認めた都度，不規則的に行われていた。

c)　年は，通常は変動する長さをもつ月の列で構成される。“うるう”の規則があるときは，年の内部構造は，一年の各月の名称と長さが規則に従って記述されたテンプレートにもとづいて，限られたパターンの中の一つに従うこととなる。“うるう”の規則は，一つの時代において特定の年の番号をテンプレートに関連づける方法について規定している。ひと月は，月の循環周期，すなわち月の満ち欠けの循環周期又は地球の周りを回転する月の循環周期と，ほぼ一致する。多くの古代の暦においては，ひと月の長さは，計算によるよりも，天体観測によって決定されることが多かった。月の周期は日の整数倍ではないし，一年は月の周期の整数倍でもない。ある種の暦は，ひと月の長さの平均値が月の周期の長さと等しくなるように暦の月の長さに制限を付けることによって，暦の月と月の周期の間で同期を保たせている。この場合，暦年は，特定の年の中に補助的な月を挿入することによって太陽の周りを回転する地球の周期と再調整される。その他の暦（例えば，グレゴリオ暦）では，月の周期と同調するようには設計されておらず，暦年は単純に整数の数の月に分割される。

d)　多くの暦では，ひと月の長さよりも短い数日の間隔を定義している。例えば，グレゴリオ暦及びその他の西洋の暦では，７日から成る一週を標準的な間隔として定義している。週の中の日は，文化的には重要な意味をもつ日となったりもするが，しかし，その日の時間位置について何らか情報が付加されるわけではない。ある祭りが特定の日というより特定の曜日に行わなければならないという要求などのために，“うるう”の規則が複雑になることがある。

e)　すべての暦は一つ以上の暦年代に関連する。暦年代は，ある種の神話的又は歴史的な事象に一致する参照日から相対的に数える，年の順列である。暦は二つのグループに分けられる。ある種の暦は，単一の参照事象から始まる相対的な年を数えるもので，一つ又は二つの暦年代をもつ。グレゴリオ暦はその実例である。さまざまな理由によって，参照事象は一つの歴史的時代から別のものへ変更することが可能である。例えば，ユリウス暦（D.3.1 参照）は幾つかの異なった参照事象とともに使われてきた。その他の暦は複数の参照事象を使用する。治世年代法（統治者の即位時点から年を数えること）は広く用いられている。現在の日本の暦（D.3.2 参照）及び古代のバビロニアの暦（D.3.3 参照）がこの方法の実例となる。

D.2　暦の記述　本体の 5.3.2 は，暦のための比較的単純なインタフェースのセットを定義し

たスキーマを記述している。そこでは暦年代の記述のための詳細な要件を規定している。ある暦日をユリウス暦日数に変換するためには，上の D.1 で議論された幾つかの要因に対して条件付けする情報が必要である。

現在も使用されているほとんどのものを含め，多くの暦では，ユリウス日付型に正確に変換するためのアルゴリズムが明確にされてきた。そのような場合には，要求される情報はアルゴリズムとして提供することができる。Dershowitz 及び Reingold（1997），Doggett（1992），Hatcher（1984，1985） 及び Richards （1998）は，いろいろなアルゴリズムを規定している。

多くの暦はほぼ規則的であり，暦による日付をユリウス日付に大体変換できるアルゴリズムを書くことができる。この場合，要求される情報はこのようなアルゴリズムとして提供することができる。この時，アルゴリズムの精度の推定が情報の中に含まれなければならず，また，その精度を維持できる期間に関する説明も含まれなければならない。

非常に不規則な暦の場合は，アルゴリズムによって適切に記述することができない。幾つかの場合は各年のパターンを記述する対照表を示すことによって可能となることがある。それは，内挿法によってほぼ正しいユリウス日付に変換することができるような，参照日の集合で，典型的な年の記述をすることだけで可能となる。Parise（1982）は膨大な暦変換表を提示している。

D.3　実例

D.3.1　ユリウス暦　ユリウス暦は規則的な暦の一例である。これは紀元前 45 年，ジュリアス・シーザーが太陽年との同期を取り戻すために古代ローマ暦から改暦することを命じた時以来，日付を刻んでいる。ユリウス暦の平年（common yaer）は 365 日の長さをもち，４年に一度，うるう年として知られている付加的な一日を含む年がある。シーザーの死後，ローマの仕官たちはその規則を誤解し，３年ごとにうるう年を設けた。紀元前 9 年に，皇帝アウグストスはこの結果累積した誤差を，紀元後 8 年までうるう年を設けないという方法によって補正することを命じた。

ユリウス暦の体系は多くの暦年代に使われてきた。詳細については，Parise（1982）及び Richards（1988）を参照せよ。 ローマの共和制後期から帝国の時代を通じて，暦年はローマの都市建設が決定されたとされる紀元前 753 年から数えられていた。西暦 525 年，Dionysius Exiguus は，イエス・キリストの誕生日とされる年から数えるよう提案した。しかしながら，このキリスト年代は 11 世紀に至るまで西ヨーロッパで広く使われることはなく，更に，ギリシャ世界においては 15 世紀になるまで受け入れられなかった。

表 D.1 － ユリウス年の各月

月の名称	平年における長さ	うるう年における長さ
January（１月）	31	31
February（２月）	28	29
March（３月）	31	31
April（４月）	30	30
May（５月）	31	31
June（６月）	30	30

July（７月）	31	31
August（８月）	31	31
September（９月）	30	30
October（１０月）	31	31
November（１１月）	30	30
December（１２月）	31	31

アウグストスの時代以来，ユリウス年は表 D.1 に示すように 12 ヶ月で構成されてきた。ローマ帝国の時代は，1 月 1 日は年の始めの日であった。後年，他の日が使われた。詳細は Parise（1982）を参照。

表 D.2 は，本体の 5.3 の要求を満たすユリウス暦の記述例である。この例は，およそ西暦 1000 年 1 月 1 日から西暦 1582 年 10 月 4 日の間キリスト年代で使われていた，1 月 1 日を年の始めとするユリウス暦を記述している。

表 D.2 － キリスト年代によるユリウス暦の記述

要　　素	値
TM_ReferenceSystem （時間参照系要素）	
TM_ReferenceSystem.name 時間参照系．名称	Julian calendar　　ユリウス暦
TM_ReferenceSystem.domainOfValidity 時間参照系．有効範囲	Western Europe　　西ヨーロッパ
TM_Calendar.dateTrans 暦要素．暦日変換 ユリウス暦日(Y/M/D)からユリウス日付(J)への変換アルゴリズム	Y′ = Y + 4716 – (14 – M) / 12 M′ = MOD(M + 9, 12) D′ = D – 1 c = (1461Y′) / 4 d = (153M′ + 2) / 5 J = c + d +D' – 1401
TM_Calendar.julTrans 暦要素．ユリウス日変換 ユリウス日付(J)からユリウス暦日(Y/M/D)への変換アルゴリズム	J′ = J + 1401 Y′ = (4J′ + 3) / 1461 T′ = MOD(4J′ +3, 1461) / 4 M' = (5T′ + 2) / 153 D' = MOD(5T′ +2, 153) / 5 D = D′ + 1 M = MOD(M′ + 2, 12) + 1 Y = Y′ - 4716 + (14 – M) / 12
Basis 基準暦年代	
TM_CalendarEra name 暦年代要素．名称	Christian era　　キリスト年代
TM_CalendarEra.referenceEvent 暦年代要素．参照事象	Birth of Jesus Christ　イエス・キリストの生誕

TM_CalendarEra.referenceDate 暦年代要素．参照日付	01，01，01
TM_CalendarEra.julianReference 暦年代要素．参照ユリウス日	1721424
TM_CalendarEra.epochOfUse.begin 暦年代要素．暦年期間．始点	2087769
TM_CalendarEra.epochOfUse.end 暦年代要素．暦年期間．終点	2299160

表 D.2 のアルゴリズムは Richards （1998）を採用した。このアルゴリズムは，他の年代によるユリウス暦や，一年の最初の日が 1 月 1 日以外で始まるユリウス暦の変換ためには修正を行う必要がある。詳細は Richards （1998）を参照せよ。すべての計算は整数演算で行われる。

D.3.2　現代日本の和暦　1873 年以来，日本はグレゴリオ暦を使っているが，天皇の在位期間に従って年を数える和暦のシステムも維持されている。表 D.3 は，この標準によって規定された要素を使った和暦の記述を例示している。

表 D.3 － 日本の暦年代

名称	参照事象	参照日付	参照ユリウス日	暦年期間 開始日	終了日
明治 (Meiji)	グレゴリオ暦の採用	明治 6.1.1	2405160	2405160	2419614
大正 (Taisho)	新しい天皇の即位	大正 1.7.31	2419615	2419615	2424875
昭和 (Showa)	新しい天皇の即位	昭和 1.12.26	2424876	2424876	2447534
平成 (Heisei)	新しい天皇の即位	平成 1.1.8	2447535	2447535	現在

D.3.3　古代バビロニア暦　古代バビロニアの暦は非常に不規則な暦の実例である。これは，一年が 12 ヶ月に区切られる太陰暦であった（表 D.4）。季節の変化と同調させるために 13 番目の月が年によって加えられた。最初（ca. 紀元前 2400 年），このうるう月は大麦（barley）の収穫が年の始めにできることを保証するために置かれた。紀元前 1000 年頃には，年の始めには月がプレアデス星団に近接するように，うるう月が加えられた。紀元前 500 年頃には，うるう年は 19 年周期とされ，Adaru II が 3,6,8,11,14 及び 19 年目に付加され，Ululu II が 17 年目に追加された。

表 D.4 － バビロニア年の各月

番号	名称	番号	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

一日は日没とともに始まった。通常，ひと月は新月が最初に出現したときに始まったので，月の長さは 29 日と 30 日の間を不規則に変動した。しかしながら，新月が観測できない場合は 30 日目が月の終わりであった。

備考　通常，現代の天文学的データから各太陰月の長さを計算することは可能であるが，暦月の長さは観測に基づいていたため，天文学上の月に相当するかどうか不明である。

年は王の治世期間を基礎として数えられた。年代の最初の年は，王位継承後の最初の年から始められた。西暦２世紀に，プロレマイオスは，紀元前 747 年までさかのぼる王とその治世期間の一覧表を作成している。この一覧表は，近代の年とバビロニア暦を関連づける際の基礎として使われてきた。他にも古代の王の一覧表が発見され，それは，初期のころの精度はあまり良くないものの，およそ紀元前 2000 年までさかのぼることができるものであった。表 D.5 はバビロニア暦の 10 年間の例である。

表 D.5 － バビロニアにおけるネブチャドネッツァ２世の治世の最初の 10 年

年	Nisanu 初日のユリウス日付	月数	日数	うるう月
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
Nebuchadnezzar10	1503470.25	12	355	

指定された月の長さは知られていないため，この暦の日付をユリウス日付に正確に変換することはできない。しかしながら，任意に与えられた年の平均的なひと月の長さにもとづくアルゴリズムを利用して正確なユリウス日を精度よく近似できる。

表 D.6 － ネブチャドネッツァ２世の時代からのバビロニア暦についての記述

要　　素	値
TM_ReferenceSystem 時間参照系要素	
TM_ReferenceSystem.name 時間参照系．名称	Babylonian calendar　　バビロニア暦
TM_ReferenceSystem.domainOfValidity 時間参照系．有効範囲	Southwest Asia　　南西アジア

TM_Calendar.dateTrans 暦要素．暦日変換 バビロニア暦日付(Y/M/D)からユリウス日付(J)への変換アルゴリズム	$J = J_Y + INT((M-1)(L/N)) + D - 1$ ただし： – 年 Y の Nisanu 初日のユリウス日付（表参照） – 月の順序数 – 年のなかの日数（表参照） – 年のなかの月数（表参照） – 月のなかの日数
TM_Calendar.julTrans 暦要素．ユリウス日変換 ユリウス日付(J)からバビロニア暦日付への変換アルゴリズム	Y = 表より，Nisanu 初日のユリウス日付(J_Y)が J より小さな最大値であるような年の値 $M = INT((J - J_Y + 1)/(L/N))$ $D = J - J_Y + 1 - INT(M*(L/M))$
Basis 基準暦年代	
TM_CalendarEra.name 暦年代要素．名称	Nebuchadnezzar II　　ネブチャドネッツァ２世
TM_CalendarEra.referenceEvent 暦年代要素．参照事象	Ascension of Nebuchadnezzar II to the throne of Babylon ネブチャドネッツァ２世のバビロニア王位継承
TM_CalendarEra.referenceDate 暦年代要素．参照日付	01, 01, 01
TM_CalendarEra.JulianReference 暦年代要素．参照ユリウス日	1721423.25
TM_CalendarEra.epochOfUse.begin 暦年代要素．暦年期間．始点	2087769
TM_CalendarEra.epochOfUse.end 暦年代要素．暦年期間．終点	2299160

D.3.4　GPS 暦　GPS（Global Positioning System）は GPS 関連データの時間位置を記述するために特定の暦法・時法を用いている。日付は 1980 年 1 月 6 日真夜中（00:00:00 UTC）を原点にもち暦年代で数えられる週数（WN）によって識別され，日は週の中の日数（DN）によって識別される。双方とも 0 から数え始められる。

表 D.7 － GPS 暦の記述

要　　素	値
TM_ReferenceSystem 時間参照系要素	
TM_ReferenceSystem.name 時間参照系要素．名称	Global Positioning System calendar　　GPS 暦
TM_ReferenceSystem.domainOfValidity 時間参照系要素．有効範囲	global　　地球全域

TM_Calendar.dateTrans 時間参照系要素．日付変換 暦日(WN,DN)からユリウス日付（J）に変換するアルゴリズム	J=2444244.5 + (7 * WN) +DN
TM_Calendar.julTrans 時間参照系要素．ユリウス日変換 ユリウス日付(J) から暦日(WN,DN)に変換するアルゴリズム	DL = J – 2444244.5 WN = INT (DL/7) DN = MOD (DL, 7)
Basis 基準暦年代要素	
TM_CalendarEra.name 暦年代要素．名称	GPS era　　GPS 暦
TM_CalendarEra.referenceEvent 暦年代要素．参照事象	
TM_CalendarEra.referenceDate 暦年代要素．参照日付	0001, 01
TM_CalendarEra.julianReference 暦年代要素．参照ユリウス日	2444244.5
TM_CalendarEra.epochOfUse.begin 暦年代要素．暦年期間．始点	2444244.5
TM_CalendarEra.epochOfUse.end 暦年代要素．暦年期間．終点	current　　現在

GPS 時間系は，日数単位の時刻としてではなく，秒で表現された週単位の時刻（TOW）として規定する独特な時計である。GPS 週は一週 604,800 秒から成る。UTC 時刻が地球の回転周期長に合わせてうるう秒を周期的に加えるのに対し，GPS 時刻は連続な尺度で流れる。GPS 時刻はまた，UTC と比較して 1 ミリ秒の範囲内で変動する。その結果，GPS 時刻と UTC を互いに変換するアルゴリズムはかなり複雑なものとなるが，この例では触れない。

附属書 E（参考）　参考文献

1) Allen, J. F., Maintaining Knowledge about Temporal Intervals, Communications of the ACM, 1983, vol. 26 pp. 832-843
2) N. Dershowitz and 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, pp. 575-608
4) D. A. Hatcher, Simple formulae for Julian day numbers and calendar dates, Journal of the Royal Astronomical Society, 1984, Vol. 25, p. 53
5) D. A. Hatcher, Generalized equations for Julian day numbers and calendar dates. Journal of the Royal Astronomical Society, 1985, Vol. 26, p. 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 19104, Geographic information- Terminology
10) ISO 19107, Geographic information- Spatial schema
11) 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/
12) 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
13) F. Parise, The book of calendars, Facts on file, New York 1982
14) J. P.Parisot, Additif to the paper of H. A. Hatcher, Journal of the Royal Astronomical Society, 1986, Vol. 27, p. 506
15) E. G. Richards, Mapping Time: The calendar and its history. Oxford University Press1998