

地理情報標準プロファイル

**Japan Profile for Geographic Information Standards (JPGIS)**

**Ver. 2.0**

平成 20 年 4 月

国土交通省国土地理院

## まえがき

この標準は、国土地理院が平成 16 年度事業として実施した「地理情報標準の利用促進に関する調査研究」の成果の一部であり、平成 15 年度までに実施された国土地理院官民共同研究の成果を参考とし、地理情報規格群（地理情報に関する国際規格（ISO 19100 シリーズ）及び日本工業規格（JIS X 7100 シリーズ））の中から、最小限の部分を取り出して体系化した、地理情報標準プロファイル（JPGIS: Japan Profile for Geographic Information Standards）である。

このプロファイルは、地球上の位置と直接又は間接に関連するものに関する情報処理技術のための基本的な標準であり、河川、道路、建物、土地などに関する様々な事象を電子化するとともに、これらを伝送することを通じて各種情報処理の高度化・効率化に役立つことを目的としている。

個別の地理情報分野ごとに制定される空間データ製品仕様書は、この地理情報標準プロファイルに準拠すべきであり、必要に応じてこのプロファイルの各章に示した拡張法を適用し、地理情報規格群に規定する範囲内で、拡張した仕様とすることができる。

## 目次

1	適用範囲.....	1
2	適合性.....	1
3	引用規格.....	1
4	UMLによる表記法.....	3
4.1	表現及び表記.....	3
4.1.1	統一モデリング言語 (UML: Unified Modeling Language) の概念.....	3
4.1.2	クラス.....	3
4.1.3	属性.....	3
4.1.4	関連.....	4
4.1.5	継承.....	4
4.1.6	ステレオタイプ.....	5
4.2	パッケージの依存性.....	7
5	基本的なデータ型.....	8
5.1	データ型.....	8
5.1.1	基本データ型.....	8
5.1.2	コレクション・データ型.....	9
5.2	測定値及び測定単位.....	9
5.2.1	Measure.....	9
5.2.2	UnitOfMeasure.....	9
6	応用スキーマのための規則.....	10
6.1	一般地物モデル.....	10
6.1.1	一般地物モデルの目的.....	10
6.1.2	一般地物モデルの主要構造.....	10
6.1.3	GF_FeatureType.....	11
6.1.4	GF_PropertyType.....	11
6.1.5	GF_AttributeType.....	12
6.1.6	GF_AssociationRole.....	12
6.1.7	GF_AssociationType.....	12
6.1.8	GF_AggregationType.....	13
6.1.9	GF_InheritanceRelation.....	13
6.2	地物型の属性.....	13
6.2.1	GF_SpatialAttributeType.....	14
6.2.2	GF_TemporalAttributeType.....	14
6.2.3	GF_LocationAttributeType.....	14
6.2.4	GF_ThematicAttributeType.....	14
6.3	一般規則.....	15

6.3.1	応用スキーマの識別.....	15
6.3.2	応用スキーマの文書.....	15
6.3.3	応用スキーマと標準スキーマの統合.....	15
6.3.4	UMLによる応用スキーマの規則.....	15
6.4	空間属性のための規則.....	15
6.4.1	一般空間規則.....	15
6.4.2	空間属性.....	15
6.5	時間属性のための規則.....	16
6.5.1	一般時間規則.....	16
6.5.2	時間属性.....	16
6.6	主題属性のための規則.....	17
6.6.1	一般規則.....	17
6.6.2	主題属性.....	17
6.7	被覆のための規則.....	17
6.7.1	一般規則.....	17
6.7.2	被覆の表現.....	17
6.8	地理識別子を用いた空間参照のための規則.....	17
6.8.1	地理識別子を用いた空間参照の表現.....	17
6.9	カタログ化規則.....	17
6.9.1	地物カタログに基づく応用スキーマ.....	17
6.10	応用スキーマのための規則の拡張及び制限のための規則.....	18
6.10.1	一般.....	18
6.10.2	拡張のための規則.....	18
6.10.3	制限のための規則.....	18
7	空間スキーマ.....	19
7.1	スキーマの構造.....	19
7.2	幾何スキーマ.....	20
7.2.1	Geometry root パッケージ.....	20
7.2.2	Geometric primitive パッケージ.....	21
7.2.3	Coordinate geometry パッケージ.....	26
7.2.4	Geometric complex パッケージ.....	33
7.2.5	Geometric aggregates パッケージ.....	34
7.3	位相スキーマ.....	35
7.3.1	一般.....	35
7.3.2	Topology root パッケージ.....	35
7.3.3	Topological primitive パッケージ.....	35
7.3.4	Topological complex パッケージ.....	39
7.4	幾何実現.....	39
7.4.1	TP_Primitive の幾何実現.....	39

7.4.2	TP_Complex の幾何実現	40
7.5	座標参照系との関連	40
7.5.1	一般	40
7.5.2	直接位置と座標参照	40
7.5.3	幾何オブジェクトと座標参照	41
7.6	空間スキーマプロファイル拡張及び制限のための規則	41
7.6.1	一般	41
7.6.2	拡張のための規則	41
7.6.3	制限のための規則	42
8	時間スキーマ	43
8.1	スキーマの構造	43
8.1.1	次元としての時間	43
8.1.2	時間のオブジェクト	44
8.2	幾何スキーマ	44
8.2.1	TM_GeometricPrimitive	44
8.2.2	TM_Instant	44
8.2.3	TM_Period	45
8.3	位相スキーマ	45
8.3.1	一般	45
8.3.2	TM_TopologicalPrimitive	45
8.3.3	TM_Node	46
8.3.4	TM_Edge	46
8.4	幾何実現	46
8.4.1	一般	46
8.4.2	TM_Node の幾何実現	47
8.4.3	TM_Edge の幾何実現	47
8.5	時間参照系との関連	47
8.5.1	時間参照系	47
8.5.2	時間位置	47
8.5.3	TM_Position	47
8.5.4	TM_TemporalPosition	48
8.5.5	暦日	49
8.5.6	時刻を伴った日付	49
8.6	時間スキーマプロファイル拡張及び制限のための規則	50
8.6.1	一般	50
8.6.2	拡張のための規則	50
8.6.3	制限のための規則	50
9	被覆の幾何及び関数のためのスキーマ	51
9.1	スキーマの構造	51

9.2	被覆基底パッケージ	51
9.2.1	CV_Coverage	52
9.2.2	CV_DiscreteCoverage	53
9.2.3	CV_GeometryValuePair	53
9.2.4	CV_ContinuousCoverage	53
9.2.5	CV_ValueObject	53
9.3	離散被覆パッケージ	54
9.3.1	CV_DiscretePointCoverage	54
9.3.2	CV_PointValuePair	55
9.3.3	CV_DiscreteCurveCoverage	55
9.3.4	CV_CurveValuePair	55
9.3.5	CV_DiscreteSurfaceCoverage	55
9.3.6	CV_SurfaceValuePair	55
9.3.7	CV_DiscreteGridPointCoverage	56
9.3.8	CV_Grid	57
9.3.9	CV_GridValuesMatrix	58
9.3.10	CV_SequenceRule	58
9.3.11	CV_GridPointValuePair	58
9.4	ティーセン多角形被覆パッケージ	59
9.4.1	CV_ThiessenPolygonCoverage	59
9.4.2	CV_ThiessenValuePolygon	60
9.5	不規則三角網被覆パッケージ	60
9.5.1	CV_TINCoverage	61
9.5.2	CV_ValueTriangle	61
9.6	空間スキーマとの関連	62
9.7	被覆の幾何及び関数のためのスキーマプロファイル拡張及び制限のための規則	62
9.7.1	一般	62
9.7.2	拡張のための規則	62
9.7.3	制限のための規則	62
10	地理識別子による空間参照	64
10.1	地理識別子による空間参照の概念	64
10.1.1	地理識別子による空間参照	64
10.1.2	地理識別子による空間参照系	64
10.1.3	地名辞典	64
10.2	地理識別子による空間参照スキーマ	64
10.2.1	SI_LocationType	65
10.2.2	SI_Gazetteer	66
10.2.3	SI_LocationInstance	66
10.3	地理識別子による空間参照プロファイル拡張及び制限のための規則	67

10.3.1	一般	67
10.3.2	拡張のための規則	67
10.3.3	制限のための規則	67
11	地物カタログ化法	68
11.1	一般	68
11.2	主要な要件	68
11.2.1	地物カタログ	68
11.2.2	カタログ要素	68
11.3	地物カタログのテンプレート	69
11.3.1	地物カタログの概念モデル	69
11.3.2	地物カタログのテンプレート	70
12	符号化	74
12.1	基本的な概念	74
12.1.1	一般	74
12.1.2	データ交換	74
12.2	文字レパートリ	76
12.3	符号化規則	76
12.3.1	概説	76
12.3.2	符号化要件概要	76
12.3.3	入力データ構造	77
12.3.4	出力データ構造	77
12.3.5	変換規則	77
12.3.6	附属書について	78
<b>附属書 1 (規定) 抽象試験項目群</b>		<b>79</b>
<b>附属書 2 (規定) 参照系</b>		<b>91</b>
<b>附属書 3 (規定) 品質</b>		<b>93</b>
<b>附属書 4 (規定) メタデータ</b>		<b>94</b>
<b>附属書 5 (規定) 定義</b>		<b>95</b>
<b>附属書 6 (規定) 他の規格から引用するクラスの定義</b>		<b>107</b>
<b>附属書 7 (参考) 参照モデル</b>		<b>116</b>
<b>附属書 8 (参考) XML に基づく符号化規則</b>		<b>119</b>
<b>附属書 9 (参考) 四辺形グリッド被覆</b>		<b>144</b>
<b>附属書 10 (参考) 描画法</b>		<b>148</b>
<b>附属書 11 (規定) 空間データ製品仕様書</b>		<b>158</b>
<b>附属書 12 (規定) 地理マーク付け言語 (GML)</b>		<b>161</b>

(参考資料)

ISO 19136 Geographic Information — Geography Markup Language (GML)

附属書 E(規定)の概要..... 164



# 地理情報標準プロファイル

## Japan Profile for Geographic Information Standards

**序文** このプロファイルは、日本工業規格、ISO 19100 シリーズで規定する応用スキーマのための規則 (ISO 19109) , 空間スキーマ (JIS X 7107) , 時間スキーマ (JIS X 7108) , 被覆の幾何及び関数のためのスキーマ (ISO 19123) , 地理識別子による空間参照 (JIS X 7112) , 地物カタログ化法 (ISO 19110) 並びに符号化 (ISO 19118) について、地理情報の概念スキーマ及び符号化規則を記述するために必要となる基本的な要素を抽出しプロファイルとして規定したものである。このプロファイルで定義するスキーマは、UML (ISO/IEC 19501) によって記述する。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るが、このプロファイルでは属性及び関連のみを扱う。

### 1 適用範囲

この地理情報標準プロファイルは、地球の位置に直接又は間接に関連付けられたオブジェクト又は現象に係る情報のための構造化された規格群のうち、異なる使用者間、システム間、場所間において地理情報を電子形式で管理、取得、呼出し、表現、転送するための手法、手段及びサービスを示すために最低限必要となる基本的な要素を示す。

このプロファイルは、可能な限り適切な情報技術及びデータの規格を参照しており、個別の地理情報の分野ごとのアプリケーションを開発するための枠組みを提供する。また、用途に応じ、各要素の拡張規則を遵守した上でこのプロファイルの拡張プロファイルを作成し、使用することができる。

### 2 適合性

このプロファイルに適合するためには、附属書 1 で規定する抽象試験項目群の要件を満たさなければならない。

また、このプロファイルで定義されていない要素を使用したい場合は、原規格を引用し、各項目の拡張規則に従って作成しなければならない。

### 3 引用規格

次に掲げる規格は、このプロファイルに引用されることによって、このプロファイルの一部を構成する。

JIS X 0301:2002 情報交換のためのデータ要素及び交換形式－日付及び時刻の表記

JIS X 4159:2005 拡張可能なマーク付け言語 (XML)

JIS X 7105:2001 適合性及び試験

JIS X 7107:2005 空間スキーマ

JIS X 7108:2004 時間スキーマ

JIS X 7111:2004 座標による空間参照

JIS X 7112:2006 地理識別子による空間参照

JIS X 7115:2005 メタデータ

日本メタデータプロファイル JMP2.0:2004

ISO/TS 19103:2005 Geographic Information – Conceptual Schema Language

ISO 19109:2005 Geographic Information – Rules for Application Schema  
ISO 19110:2005 Geographic Information – Methodology for Feature Cataloguing  
ISO 19117:2005 Geographic Information – Portrayal  
ISO 19118:2005 Geographic Information – Encoding  
ISO 19123:2005 Geographic Information – Schema for Coverage Geometry and Function  
ISO 19131:2007 Geographic Information – Data Product Specifications  
ISO 19136:2007 Geographic Information – Geography Markup Language (GML)  
ISO/IEC 19501:2005 Information technology -- Open Distributed Processing -- Unified  
Modeling Language (UML) Version 1.4.2

## 4 UML による表記法

### 4.1 表現及び表記

#### 4.1.1 統一モデリング言語 (UML : Unified Modeling Language) の概念

このプロファイルでは、概念スキーマは UML を使用して表現する。

#### 4.1.2 クラス

UML クラスには、属性、操作及び関連役割の三つの基本的な特性があるが、このプロファイルでは属性及び関連役割のみを扱う。クラスは、図 4-1 のように表記し、ステレオタイプ及びクラスの名前を一段目に記述し、二段目に属性を記述する。三段目には操作を記載するが、このプロファイルでは扱わないため、空白となる。

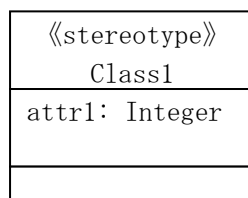


図4-1 UML におけるクラスの表記例

#### 4.1.3 属性

属性は、以下 OMG UML1.4.2 に準拠し、UML クラス図の中に表現する。属性は図 4-2 のように表記する。

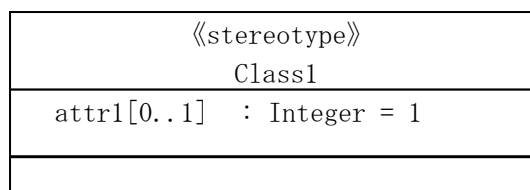


図4-2 UML における属性の表記例

属性の記載は、以下の a) から g) で規定する方法による。なお、以下の例は、図 4-2 の記載内容と対応する。

- a) **visibility (可視性)** 属性に対するオブジェクト外部からの可視性を示す。visibility の記載は省略可能であり、かつこのプロファイルでは、パブリック (+) のみを扱うため、このプロファイルにおいて visibility は省略する。

b) name (名前) 属性の名称。

例 attr1

c) multiplicity (多重度) この属性がもつことのできる値の数。[a..b]は、 $a \leq j \leq b$  となる任意の整数  $j$  を意味する。[a..a]は、[a]と同じとみなす。

例 0..1

d) begin-value (開始値) 有効な多重度を表す任意の整数。終了値が続かない場合は、開始値のみを多重度とする。

例 0

e) end-value (終了値) 先行する開始値より大きな値をとる整数。終了値が無限又は非有界な基数の範囲を表す場合は“\*”とする。

例 1

f) type (型) 先行する属性のオブジェクト又は値の型。

例 Integer

g) initial-value (初期値) 属性の初期値。

例 1

なお、派生属性の記載は、属性名にスラッシュ (/) を付ける (図 4-3)。ここで派生属性とは、自分が関係しているクラスの情報から加工・算出できる値を保有する概念的な属性である。実際に属性値をもつかどうかは符号化に委ねる。

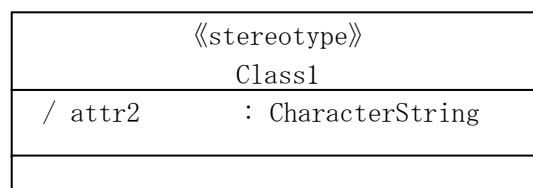


図4-3 UML における派生属性の表記例

#### 4.1.4 関連

関連には関連名、関連の対象となる各クラスには役割名を付けることができる。図 4-4 の図式の場合、“Class1” クラスは “Class2” クラスと “Relation” という名前で関連し、この二つのクラスを実装する場合には、通常この関係の役割の名前をつけた “属性” として他方のクラスへの参照をもたせる。

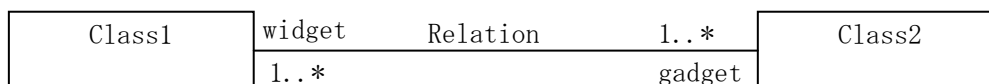


図4-4 UML における関連の表記例

#### 4.1.5 継承

継承とは、既に定義されているクラス (上位クラス) をもとに、属性や関連を加えた新しいクラス (下位クラス) を定義することである。上位クラスの特長 (属性や関連役割) は下位クラス

にも受け継がれる。図 4-5の図式の場合，“Class1”クラスに定義された属性“attr1”を，“Class1”クラスを継承する“Class2”クラスももつ。

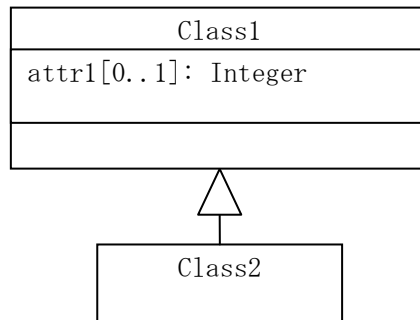


図 4-5 UML における継承の表記例

また，上位クラスの特長（属性や関連役割）を下位クラスが置き換えることもできる。これをオーバーライドと呼ぶ。図 4-6の図式の場合，“Class1”クラスに定義された属性“attr1”を，“Class2”クラスはオーバーライドしている。“Class1”クラスでは属性の多重度が[0..1]であるが，“Class2”クラスでは，多重度が[1]にオーバーライドする。

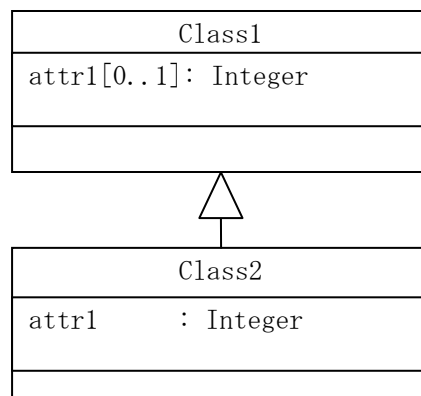


図4-6 オーバーライドの表記例

#### 4.1.6 ステレオタイプ

UML のステレオタイプは，既存の UML の概念を拡張する仕組みとする。それはモデル要素であり，既存の基本メタモデルクラスを基礎として作り，あたかも新たな仮想的又は擬似的なメタモデルクラスであるかのように振る舞うものとして，別の UML 要素と区別（意味付け）するために使用する。ステレオタイプは，本来備わった UML メタモデルクラスの階層に基づく分類の仕組みを拡張する。このプロファイルで使用するステレオタイプを以下に示す。

##### 4.1.6.1 Abstract

直接インスタンス化できないクラス（又は他の分類を指す言葉）に適用するステレオタイプ。その名前を示す UML の表記にはイタリック体を使用する。

##### 4.1.6.2 ApplicationSchema

応用スキーマを取り扱ったパッケージに適用するステレオタイプ。

#### 4.1.6.3 CodeList

《Enumeration》より自由な列挙を記述するために使用する符号リスト (CodeList) に適用するステレオタイプ。《CodeList》は、自由な (柔軟性のある) 列挙とする。符号リストは、とることが可能な値の長いリストを表現するのに役立つ。リストの要素が完全にわかる場合 (リストの要素以外は存在し得ないと判断できる場合) は、《Enumeration》を使用することが望ましく、起こりそうな値しかわからない場合 (リストの要素が一例でしかない場合) は、符号リストを使用することが望ましい。

#### 4.1.6.4 DataType

個々の出現ごとに区別する必要がなく、副作用のある操作 (すでにある値を変更する操作) はもたない、値の集合の記述子に適用するステレオタイプ。個々の出現ごとに区別する必要がないため、識別子をもたず、他から参照されることはない。データ型には、あらかじめ定義する型と使用者が定義できる型とがある。あらかじめ定義する型として、基本データ型がある。

#### 4.1.6.5 Enumeration

そのインスタンスが、名前付けしたリテラルの一覧を形成するデータ型に適用するステレオタイプ。列挙する名前及びリテラルの両方を宣言する。列挙 (Enumeration) は、そのクラスの値として許される、十分合意が得られた値の一覧を意味する。

#### 4.1.6.6 Feature

地物に適用するステレオタイプ。このステレオタイプをもつクラスは、応用スキーマのパッケージ内で定義される。

#### 4.1.6.7 Interface

ある要素の振る舞いを特徴付けるような、名前の付いた操作の集合に適用するステレオタイプ。

#### 4.1.6.8 MetaClass

インスタンスがクラスになるクラスに適用するステレオタイプ。メタクラスは、通常メタモデルにおいて使用する。メタクラスは、他のクラスのメタデータを保持することを主な目的とするオブジェクトクラスとする。

#### 4.1.6.9 Leaf

いかなる下位パッケージももたない、クラスを含むパッケージに適用するステレオタイプ。

#### 4.1.6.10 Package

下位パッケージを含む、論理的に関係のあるクラスとパッケージの集まりに適用するステレオタイプ。

#### 4.1.6.11 Record

ある属性とその属性に対する値をもつクラスに適用するステレオタイプ。

#### 4.1.6.12 Type

インスタンス（オブジェクト）の定義域を，そのオブジェクトに適用できる操作と共に規定するために使用するステレオタイプ。タイプ型は，識別子をもち，他から参照することができる。また属性及び関連をもってもよい。

#### 4.1.6.13 Union

指定したいいくつかの型のうちの一つだけが選択される共用体に適用するステレオタイプ。これは，使用することのできる選択すべきクラス・型の集合を，共通の上位型・クラスを作成することなく指定するのに役立つ。

### 4.2 パッケージの依存性

このプロファイルの後続の箇条は，UML パッケージという概念でまとめている。パッケージは，しばしばシステムの他のパッケージが提供するサービスを呼び出すので，通常，完全なシステムを形成することはない。クライアントとして働くあるパッケージがサーバとして働く別のパッケージを使用して必要なサービスを供給する場合を，「クライアントパッケージは，サーバパッケージに依存する」という。この依存性は，パッケージ中のオブジェクトがサーバパッケージの定義する別のオブジェクトにアクセスするときに発生する。

このパッケージ依存性は図 4-7の図形記法を用いたパッケージ図式で表す。



図4-7 UML のパッケージ依存性の例

## 5 基本的なデータ型

**序文** 本章では、ISO/TS 19103 を参考文献とし、このプロファイルで使用するデータ型の定義を行う。

### 5.1 データ型

このプロファイルで使用するデータ型は以下のものである。

#### 5.1.1 基本データ型

基本データ型は、値を表現するための基本的な型である。このプロファイルで使用する基本データ型には以下のものがある。

##### 5.1.1.1 Number

数値型 (Number) は、抽象型であり、整数、実数及び十進数の上位型である。

##### 5.1.1.2 Integer

整数型 (Integer) は、符号付き整数である。

**例** 794    -645

##### 5.1.1.3 Decimal

十進数型 (Decimal) は、符号付き実数であり、数値を固定小数点数によって表す。

**例** 63.700    -645.00

##### 5.1.1.4 Real

実数型 (Real) は、符号付き実数であり、数値を仮数部と指数部によって表現する。字句空間では数値を仮数部と指数部で必ずしも分けて表現する必要はない。

**例** -1.086E3    63.718

##### 5.1.1.5 Vector

ベクトル型 (Vector) は、数値型 (Number) をとる値を複数列挙して表記する、数値の列である。

**例** (234, 123, 567, 398)    (63.718, -8.005, 90.902)

##### 5.1.1.6 CharacterString

文字列型 (CharacterString) は、任意の文字数の文字の連なりである。

**例** “文字列のサンプル”

##### 5.1.1.7 Date

日付型 (Date) は、暦日を示し、JIS X 0301 に基づき表記する。

**例** 2004-08-12    20040812

##### 5.1.1.8 Time

時間型 (Time) は、時刻を示し、JIS X 0301 に基づき表記する。



**例** 21:03:15 09:24:38+09:00

#### 5.1.1.9 DateTime

日時型 (DateTime) は、暦日と時刻を示し、JIS X 0301 に基づき表記する。

**例** 2004-08-12T09:24:38+09:00

#### 5.1.1.10 Boolean

真偽値型 (Boolean) は、true 又は false が指定できるブール数である。

**例** true false

### 5.1.2 コレクション・データ型

コレクション・データ型は、さまざまな型のデータが複数存在することを表現するためのデータ型である。このプロファイルではコレクション・データ型として、集合型 (Set) , 順序型 (Sequence) , 辞書型 (Dictionary) を使用する。

#### 5.1.2.1 Set<T>

集合型 (Set<T>) は、有限個数のデータの集合である。重複するインスタンスを含んではならない。また要素の順序付けの規定はない。T は、データの型となる。

**例** Set<GM\_Point> : 点要素 (GM\_Point) の集合を表す。

#### 5.1.2.2 Sequence<T>

順序型 (Sequence<T>) は、順序付けをもった有限個数のデータの集合である。重複するインスタンスを許す。T は、データの型となる。

**例** Sequence<Integer> : 整数型数値の配列を表す。

#### 5.1.2.3 Dictionary

一つの値をキーとして、別の値との対応を示す型である。

**例** Dictionary<Integer, GM\_Point> : 整数型数値が点 (GM\_Point) と対応している辞書を表す。

## 5.2 測定値及び測定単位

測定値及び測定単位を用いる場合は以下の型を使うものとする。

### 5.2.1 Measure

測定値。ある実体の範囲、次元又は量を確認するための行為若しくは処理が与える結果。一つの属性をもつ。属性 value は、UnitOfMeasure で指定する単位を用いた値を示す。長さ (Length) , 距離 (Distance) 及び時間 (Time) などがある。

### 5.2.2 UnitOfMeasure

測定単位。一つの属性 uomName をもつ。属性 uomName は単位の名称を示す。ここで、標準となる単位の例は以下のとおりである。

- ・ 長さ：メートル
- ・ 時間：秒
- ・ 角度：ラジアン
- ・ 面積：平方メートル
- ・ 体積：立方メートル
- ・ 速度：メートル毎秒

## 6 応用スキーマのための規則

**序文** このプロファイルは、ISO 19109 (Geographic information – Rules for application schema) から、応用スキーマを記述するために必要となる基本的な規則を抽出及び翻訳し、プロファイルとして定義したものである。

### 6.1 一般地物モデル

ここでは、地物を定義するために用いられる概念とこれらの概念がどのように関連するかを識別して記述する。記述は、概念モデルで表現し、これを一般地物モデル (GFM) と呼ぶ。

#### 6.1.1 一般地物モデルの目的

GFM は、現実世界の見方を分類するために必要な概念モデルである。概念モデルは概念スキーマ言語 (UML クラス図) で表現する。UML はそれ自身の概念モデルをもつ (メタモデル)。GFM 及び UML メタモデルは概念の分類を扱うので、両者は非常に類似しているが、一点だけ大きな違いがある。GFM 中の概念は地物の分類の基礎を確立するのに対して、UML メタモデルは任意のオブジェクトの分類の基礎を示す。

取り上げたい事や物を地物型と呼ぶ。地物型間の関係は、地物関連型と継承である。地物型は、地物属性及び地物関連役割の二つのプロパティをもっている。これらのすべての概念は GFM の UML メタクラスとして表現されるので、GFM は地物型のメタモデルである。

応用スキーマは概念スキーマ言語で表現しなければならない。応用スキーマは論議領域を表現するデータ集合の構造と内容を記述しなければならない。GFM は地物の分類のための要求を規定するが、概念スキーマ言語ではない。すなわち、応用スキーマを定義するためには、既存の概念スキーマ言語を使用しなければならないことを意味する。ISO19100 シリーズの標準スキーマを統合するため、応用スキーマは UML で記述するとよい。このプロファイルでは GFM の概念を UML に写像するための主な規則を定義する。

GFM は、UML で応用スキーマを作成する時に留意しなければならない地物を分類するための構造を規定する。しかし、GFM から UML への写像は一方向の写像であり、逆方向への写像は可能ではない。例えば、応用スキーマは UML クラスをもっている。これらのクラスのうち幾つかが GFM 地物型であり、いくつかが地物属性のためのデータ型である。UML ではこれらの型を区別することができない。また、GFM は地物定義の構造と内容を規定するだけで、地物属性値を規定することはない。

#### 6.1.2 一般地物モデルの主要構造

図 6-1は、地物型を規定するために用いられる概念を示す。

名前と記述以外に、地物型は以下のプロパティで規定する。

- 地物属性
  - 地物型を特徴づける地物関連役割
- 付加的な概念は、以下のとおりである。
- 他の地物型への汎化又は特化関係

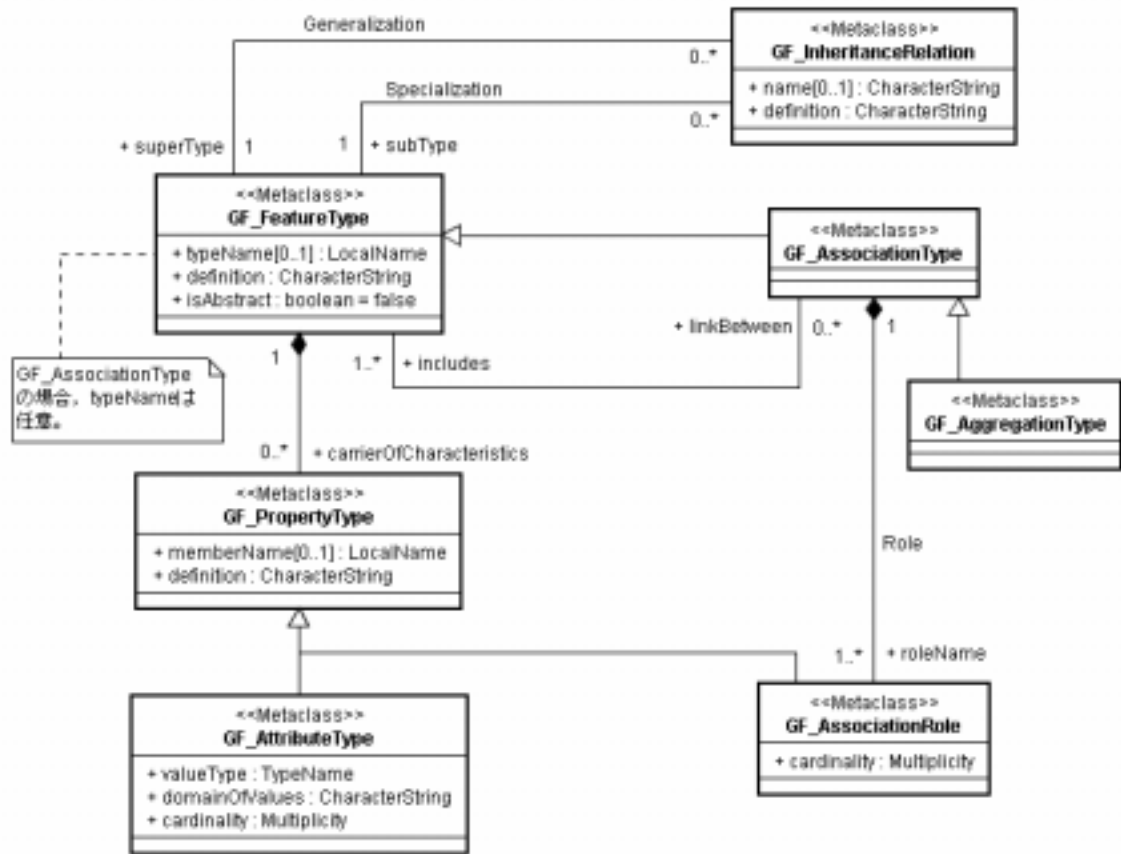


図6-1 一般地物モデル

### 6.1.3 GF\_FeatureType

地物は実世界の現象の抽象概念である。GF\_FeatureType は、個々の地物型を表すクラスとしてインスタンス化されるメタクラスである。ある地物型は、その地物型のすべてのインスタンスのためのクラスである。個々の地物型を表現しているクラスのインスタンスは、地物インスタンスである。

**備考** オブジェクト指向モデリングにおいて、地物型はクラスと等価であり、地物インスタンスはオブジェクトと等価である。

#### 属性

1. `typeName:CharacterString` は、地物型の名前を示す。名前は応用スキーマ内で一意でなければならない。
2. `definition:CharacterString` は、地物型を記述する定義を示す。
3. `isAbstract:Boolean=false` は真偽値で、もし真ならば地物型は抽象上位型として振る舞う。

#### 関連

1. `carrierOfCharacteristics` 関連役割は、地物型の特性となる地物属性型及び地物関連役割を規定する。
2. `Generalization` 関連は、地物属性や地物関連役割等の地物特性を継承する上位型を規定する。
3. `Specialization` 関連は、地物属性や地物関連役割等の地物特性を継承させる下位型を規定する。

### 6.1.4 GF\_PropertyType

GF\_PropertyType は、地物の特性及び地物の関連役割を表す、すべての地物型のプロパティのメタクラスである。

GF\_PropertyType は、GF\_AttributeType 及び GF\_AssociationRole の上位型である。

#### 属性

1. memberName:CharacterString は、属性又は関連役割の名前を示す。
2. definition:CharacterString は、地物型の属性又は関連役割の記述を示す。

#### 関連

1. featureType 関連役割は、地物特性を含む地物型を規定する。

#### 6.1.5 GF\_AttributeType

GF\_AttributeType は地物型の属性定義のためのメタクラスである。

#### 属性

1. valueType:CharacterString は、属性の型を示す。  
**例** 整数、文字列又は GM\_Object。
2. domainOfValues:CharacterString は、値のセットの記述を示す。  
**例** 正の数、3 から 7 までの値
3. cardinality:Multiplicity は、ある範囲で提供される地物型の属性と関連する値の数を示す。
4. [上位型から継承する属性]memberName:CharacterString は、属性の名前を示す。
5. [上位型から継承する属性]definition:CharacterString は、属性の記述を示す。

#### 関連

1. [上位型から継承する関連]featureType 関連役割は、地物属性を含む地物型を規定する。

#### 6.1.6 GF\_AssociationRole

GF\_AssociationRole は、役割のためのメタクラスである。

**備考** GF\_AssociationRole は、関連における地物型の前後関係及びインスタンスを特定する、地物型が振る舞う役割を示す。また、ある一つの地物型のための役割を与える GF\_AssociationRole のインスタンスは、この地物型の一部と考えられ、カプセル化される。

#### 属性

1. cardinality:Multiplicity は、この役割で振る舞うことができ、もう一方の関連終端にある地物型の単一インスタンスと関連する地物型のインスタンスの数を示す。
2. [上位型から継承する属性]memberName:CharacterString は、関連役割の名前を示す。
3. [上位型から継承する属性]definition:CharacterString は、関連役割の記述を示す。

#### 関連

1. Role 関連は、地物関連役割が部分となる関連を規定する。
2. [上位型から継承する関連]featureType 関連役割は、地物関連役割を含む地物型を規定する。

#### 6.1.7 GF\_AssociationType

GF\_AssociationType は、地物型の間に関連を示す。GF\_AssociationType は GF\_FeatureType の下位型であり、JPGIS では、GF\_FeatureType から二つの属性を継承する。

#### 属性

1. [上位型から継承する属性]typeName[0..1]:CharacterString は、地物関連の名前を示す。名前は応用スキーマ内で一意でなければならない。ただし、地物関連では、この属性は任意である。

2. [上位型から継承する属性]definition:CharacterString は、地物関連を記述する定義を示す。

#### **関連**

1. roleName 関連役割は、地物関連と、それに割り当てられる特定の関連役割を関連付ける。

#### **6.1.8 GF\_AggregationType**

GF\_AggregationType は GF\_AssociationType の下位型であり、地物型の間に関連の中で、集成と合成の関連を示す。

#### **6.1.9 GF\_InheritanceRelation**

GF\_InheritanceRelation は、より一般的な地物型（上位型）と一つ以上の特化した地物型（下位型）の間の一般的な関係を表すクラスである。

特化した地物型のインスタンスであればどれでも、上位型のインスタンスにもなる。個々の特化には目的があり、それぞれ異なる目的をもって定義される汎化階層中の地物型の中で、それ自体が上位型となることがある。

#### **関連**

1. supertype 関連役割は、一方の又は他の地物型の、より一般的な地物型であることを示す。
2. subtype 関連役割は、一方の又は他の地物型の、より特化した地物型であることを示す。

### **6.2 地物型の属性**

ここでは、地物属性の役割をより詳細に記述する。属性型 (GF\_AttributeType) は、name, description, valueType, domainOfValues 及び cardinality をもつ。また、attributeOfAttribute 関連は、ある属性型を、この属性型の性質を記述する別の属性型と関連付ける。

属性はすべて、地物の静的な情報をもつ。これは、空間及び非空間の特性を扱う。このプロファイルには、特有の属性型が存在する。これらは、図 6-2において GF\_AttributeType の下位型として示す。属性は、このプロファイルに含まれる様々なスキーマを使用することにより、インタフェースとなる。属性型は、これらのスキーマから値の型定義とスキーマに従った値の定義域を取得する。例えば、空間属性型 (GF\_SpatialAttributeType) は、このプロファイルの空間スキーマで記述された GM\_Object 又は TP\_Object の定義に従った値の型と値の定義域をもつ。

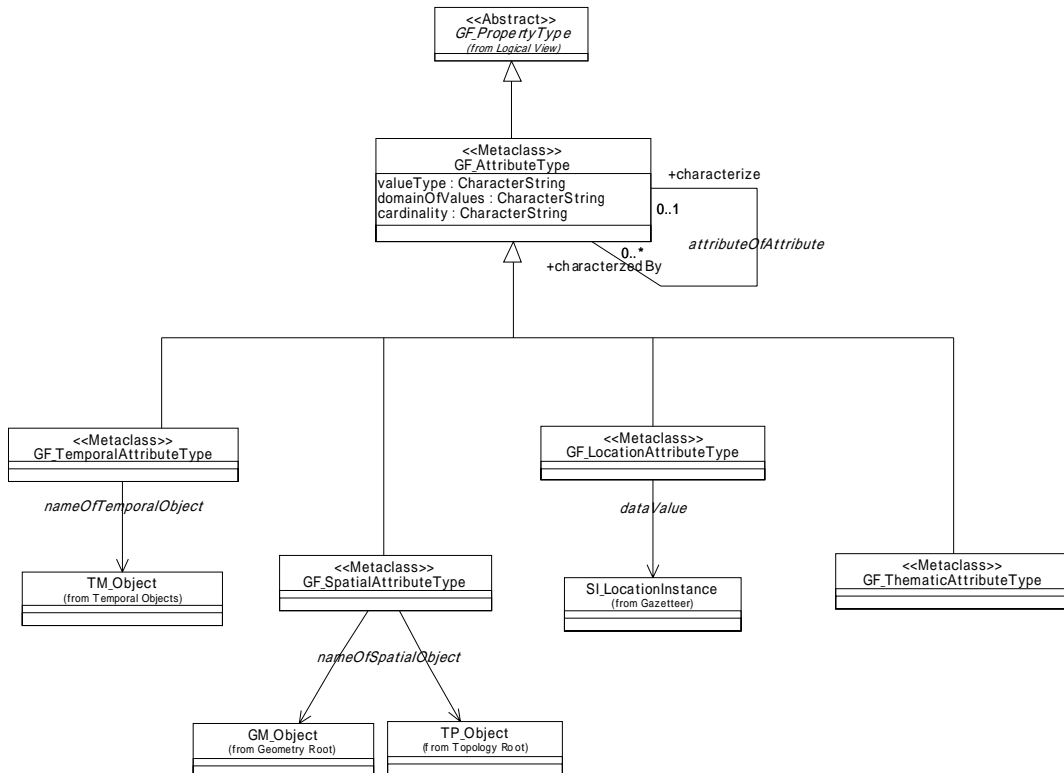


図6-2 地物型の属性

### 6.2.1 GF\_SpatialAttributeType

GF\_SpatialAttributeType は、地物型の空間的な特性を表現するために用いる空間属性を表す。空間の属性型は値の型として GM\_Object 又は TP\_Object をもたなければならない。GM\_Object 及び TP\_Object の構造は、このプロファイルの7. 空間スキーマで規定する。

### 6.2.2 GF\_TemporalAttributeType

GF\_TemporalAttributeType は、地物型に特有な時間参照として用いる時間的な属性を表す。時間的な属性型は値の型として TM\_Object をもたなければならない。TM\_Object の構造は、このプロファイルの8. 時間スキーマで規定する。

### 6.2.3 GF\_LocationAttributeType

GF\_LocationAttributeType は、地理識別子による地物の空間参照をもつ属性を表す。その構造は、このプロファイルの 10. 地理識別子による空間参照で規定する。

**例** 郵便配達区域はコードによって識別されるかもしれないが、その位置は地名辞典で見出される。これらの属性型は、このプロファイルの地理識別子による空間参照で規定する SI\_LocationInstance によって規定された値の型をもたなければならない。

### 6.2.4 GF\_ThematicAttributeType

GF\_ThematicAttributeType は、6.2.1から6.2.3で規定されるものを除いた地物の他の記述的な特性をもつ属性を表す。これらの値の型と値の定義域は、通常、使用者又は応用システム領域で規定する。

## 6.3 一般規則

### 6.3.1 応用スキーマの識別

#### 規則

1. 各応用スキーマの識別情報は、名前と版からなる。版を含むことにより、供給者と使用者が、特定のデータ集合の内容を記述する応用スキーマの版について相互に確認することができる。
2. UML では、応用スキーマはパッケージの中に記述されなければならない、応用スキーマの名前と版がパッケージの文書として記述されなければならない。

### 6.3.2 応用スキーマの文書

#### 規則

1. 応用スキーマは文書化されなければならない。
2. UML で記述する応用スキーマの文書化には、応用スキーマを作るにあたって使用するソフトウェアの中に文書化ツールがあり、出力可能であれば、それを使用することができる。
3. クラス又は他の UML 構成要素が地物カタログ情報に一致するならば、カタログへの参照が明記されなければいけない。
4. 応用スキーマの中の地物型の文書は、11. 地物カタログ化法に従わなければならない。

### 6.3.3 応用スキーマと標準スキーマの統合

#### 規則

1. 応用システムのデータ構造は、応用スキーマにおいてモデル化されなければならない。
2. データ構造の完全な定義を作り上げ、標準スキーマ（空間スキーマ、時間スキーマ、被覆の幾何及び関数のためのスキーマ、地理識別子による空間参照）と応用スキーマの統合を記述するためには、UML の依存の仕組みを使用しなければならない。
3. データ交換のために応用スキーマの中で使用される全てのクラスは、インスタンス化可能でなければならない。

**備考** 抽象クラスは、インスタンス化されることはないが、その性質を受け継ぐ下位の具象クラスがインスタンス化される。

### 6.3.4 UML による応用スキーマの規則

#### 規則

1. GF\_FeatureType のインスタンスは、クラスとして実装されなければならない。
2. GF\_AttributeType のインスタンスは、属性として実装されなければならない。
3. GF\_AssociationRole のインスタンスは、関連役割名として実装されなければならない。

## 6.4 空間属性のための規則

### 6.4.1 一般空間規則

#### 規則

1. 空間属性型の値の定義域は、地物の空間特性を記述するための概念スキーマである7. 空間スキーマによって与えられる仕様に合致していなければならない。

### 6.4.2 空間属性

#### 規則

1. 地物の空間特性は、一つ以上の空間属性によって記述しなければならない。応用スキーマの

中では、空間属性は地物属性の下位型であり、その分類法はこのプロファイルの空間スキーマで規定する。

2. 応用スキーマにおいて、空間属性は次に示す二つの方法のいずれかで表現しなければならない。
  - a) 地物を表す UML クラスの属性として定義する方法。このとき属性は、このプロファイルの空間スキーマにおいて規定する空間オブジェクトの一つをその値のデータ型として使用しなければならない。
  - b) 地物を表すクラスと、このプロファイルの空間スキーマにおいて規定する空間オブジェクトの一つとの間の UML 関連として定義する方法。
3. 空間属性は、その値として空間オブジェクトを採用しなければならない。空間オブジェクトは幾何オブジェクト又は位相オブジェクトに分類され、いずれもプリミティブ、複体又は集成（ただし集成は幾何オブジェクトのみ）として下位型化される。表 6-1は、空間属性の値として応用スキーマにおいて、以下の空間オブジェクトしか使用してはならない。

表6-1 応用スキーマにおいて空間属性として有効な空間オブジェクト一覧

幾何オブジェクト			位相オブジェクト	
幾何プリミティブ	幾何複体	幾何集成	位相プリミティブ	位相複体
GM_Point GM_Curve GM_Surface	GM_CompositeCurve GM_Complex	GM_Aggregate	TP_Node TP_Edge TP_Face TP_DirectedNode TP_DirectedEdge TP_DirectedFace	TP_Complex

## 6.5 時間属性のための規則

### 6.5.1 一般時間規則

#### 規則

1. 地物に適用される時間的側面のいかなる記述もこのプロファイルの時間スキーマで規定する仕様に合致していなければならない。

### 6.5.2 時間属性

#### 規則

1. 地物の時間特性は、地物属性の下位型である時間属性として定義しなければならない。
2. UML を用いた時間属性の実装は、標準スキーマを参照するための規則（6.3.3参照）に従わなければならない。
3. 時間属性は、地物を表現する UML クラスの属性として応用スキーマの中で表現しなければならない。その場合には、属性はその値のためのデータ型としてこのプロファイルの時間スキーマ参照の中で規定する時間オブジェクトの一つをもたなければならない。表 6-2は、時間属性の値として応用スキーマにおいて、以下の時間オブジェクトしか使用してはならない。

表6-2 応用スキーマにおいて時間属性として有効な時間オブジェクト一覧

時間幾何プリミティブ	時間位相プリミティブ
TM_Instant TM_Period	TM_Node TM_Edge



## 6.6 主題属性のための規則

### 6.6.1 一般規則

#### 規則

1. 地物に適用される主題的性質の記述は全て、このプロファイルで提供される仕様に合致していなければならない。

### 6.6.2 主題属性

#### 規則

1. 地物の主題特性は、地物属性型の下位型である主題属性として定義しなければならない。
2. 主題属性は、地物を表現する UML クラスの属性として応用スキーマの中で表現しなければならない。その場合には、属性はその値のためのデータ型としてこのプロファイルの基本データ型の中で定義されたデータ型の一つをもっていなければならない。

## 6.7 被覆のための規則

### 6.7.1 一般規則

#### 規則

1. 地物に適用される被覆要素のいかなる記述もこのプロファイルで規定する仕様に合致していなければならない。

### 6.7.2 被覆の表現

#### 規則

1. 地物型が被覆の場合は、このプロファイルの被覆の幾何及び関数のためのスキーマで規定する被覆型の一つの下位型として記述しなければならない。
2. UML を用いた被覆型の実装は、標準スキーマを参照するための規則（6.3.3参照）に従わなければならない。
3. 被覆型は、地物を表現する UML クラスの下位型として応用スキーマの中で表現しなければならない。

## 6.8 地理識別子を用いた空間参照のための規則

### 6.8.1 地理識別子を用いた空間参照の表現

#### 規則

1. 地理識別子によって空間参照される属性の値の定義域は、このプロファイルの地理識別子による空間参照において規定する仕様に従わなければならない。
2. 地理識別子は、空間参照の値をもつ属性（GF\_LocationAttributeType）によって、応用スキーマから参照しなければならない。
3. GF\_LocationAttributeType は、応用スキーマにおいて、地物を表現する UML クラスの属性として表現しなければならない。その場合には、属性はその値のためにデータ型として地名辞典スキーマ（10. 地理識別子による空間参照を参照）の中で規定する SI\_LocationInstance を使用しなければならない。

## 6.9 カタログ化規則

### 6.9.1 地物カタログに基づく応用スキーマ

## 規則

1. このプロファイルに基づく地物カタログの情報は、6.1及び6.2の規定を使用することにより、応用スキーマを作成するために使用することができる。

## 6.10 応用スキーマのための規則の拡張及び制限のための規則

### 6.10.1 一般

応用スキーマのための規則を拡張及び制限する場合の規則を定める。プロファイルを拡張して用いる場合は抽象試験項目群を設定し、適合性試験に合格しなくてはならない。

### 6.10.2 拡張のための規則

このプロファイルで規定する応用スキーマのための規則を拡張して使用することができる。

#### 規則

1. このプロファイルで定義されておらず、応用スキーマのための規則に定義されている一般地物モデルの要素を用いたい場合は、原規格から必要な要素を抽出し、原規格の記述規則に従ってプロファイルに追加定義する。

**例** 地物操作をモデルに定義する場合は、地物操作を示す `GF_Operation` を一般地物モデルに定義する。この場合は、原規格に基づき、`GF_PropertyType` から継承する。

2. 追加するクラスが必須の属性及び関連をもつ場合は、もれなく追加する。

### 6.10.3 制限のための規則

このプロファイルで定義された応用スキーマのための規則を制限し、その一部を使用することができる。

#### 規則

1. 応用スキーマのための規則の一部を制限して使用する場合は、プロファイルの修正は行わず、応用スキーマに記載することによって制限する。

**例** 本プロファイルの一般地物モデルのうち、`GF_AssociationRole` を用いないときは、プロファイルを修正し `GF_AssociationRole` を削除するのではなく、応用スキーマにおいて地物関連役割を用いないことを記述する。

## 7 空間スキーマ

**序文** このプロファイルは、ISO 19107 (Geographic information – Spatial schema) を翻訳し、技術的内容及び規格票の様式を変更することなく作成した JIS X 7107 (地理情報—空間スキーマ) から、空間属性を記述するために必要となる基本的な要素を抽出し、プロファイルとして規定したものである。

### 7.1 スキーマの構造

ここでは、地理情報の空間特性を記述する概念スキーマを示す。このスキーマは、OMG UML1.4.2 によって記述する。このスキーマは抽象モデルであり、ある実装がこのプロファイルに適合するためには、抽象モデルにおけるそれぞれのパッケージ、型及び関連で記述した内容を実現することができなくてはならないが、同じ方法で実装する必要はない。

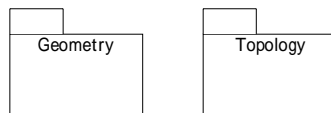


図7-1 空間スキーマパッケージ

このスキーマは、二つのパッケージからなる (図 7-1参照)。Geometry パッケージは、座標幾何、幾何プリミティブ、幾何複体及び幾何集成体を示す内部パッケージをもち、Topology パッケージは位相プリミティブ及び位相複体を示す内部パッケージをもつ (図 7-2参照)。これらのパッケージに定義された型は、地物とデータ集合との空間特性の値として使わなければならない。

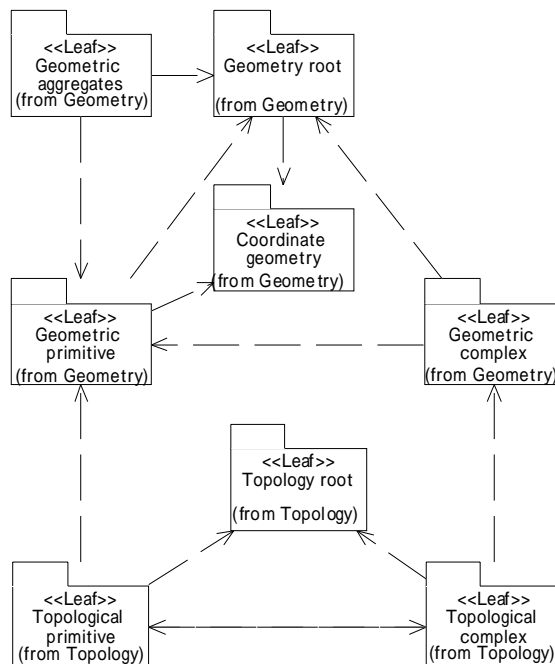


図7-2 規定の UML パッケージの依存性

このプロファイルで定義する UML クラス名は、特定のパッケージを識別するため、二文字の英字及び一文字の下線からなる接頭辞から開始する。空間スキーマで定義する型を識別する接頭辞は、幾何オブジェクトでは GM\_、位相オブジェクトでは TP\_とする。

## 7.2 幾何スキーマ

### 7.2.1 Geometry root パッケージ

Geometry root（幾何基底）パッケージには、全ての幾何オブジェクトの基底型となる GM\_Object が定義される（図 7-3 参照）。

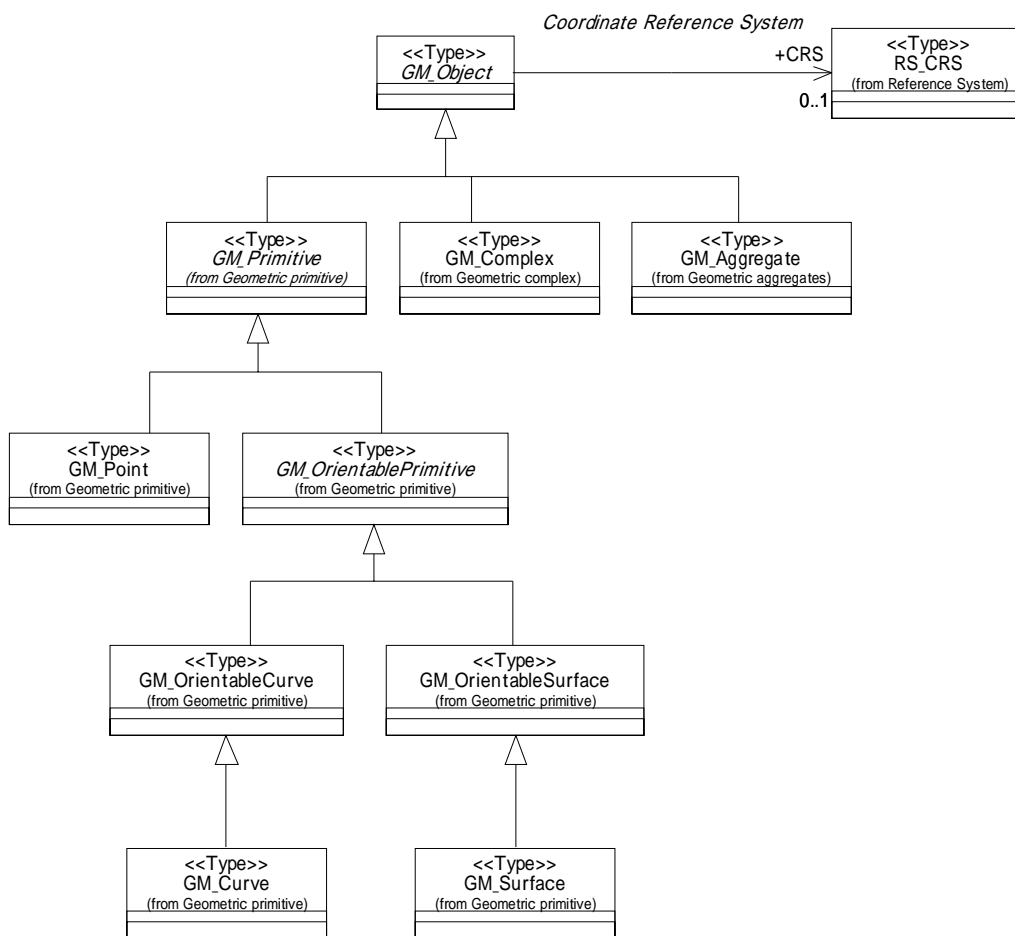


図7-3 GM\_Object の下位クラス

#### 7.2.1.1 GM\_Object

GM\_Object は、幾何オブジェクトの基底型である。

#### 関連

1. Coordinate Reference System 関連によって、この GM\_Object と DirectPosition の座標に用いる座標参照系とをつなげる。この関連は多重度が [0..1] であり、0 の場合はこの

GM\_Object はそれを包含する他の GM\_Object の RS\_CRS を用いる。座標参照系の役割名は CRS である。

## 7.2.2 Geometric primitive パッケージ

Geometric primitive (幾何プリミティブ) パッケージは、あらゆる幾何プリミティブとそれらの境界を記述するために必要となるデータの型を定義する。

### 7.2.2.1 GM\_Boundary

幾何境界型である GM\_Boundary (図 7-4) は、幾何オブジェクトの境界を表すデータに用いる全ての型の抽象基底型である。なお、境界オブジェクトは輪体となる。

### 7.2.2.2 GM\_PrimitiveBoundary

幾何プリミティブ境界型である GM\_PrimitiveBoundary (図 7-4) は、GM\_Primitive の下位型の境界を示す型である。

### 7.2.2.3 GM\_SurfaceBoundary

曲面境界型である GM\_SurfaceBoundary (図 7-4) は、GM\_Surface の境界を示す型である。

#### 関連

1. 役割名 exterior によって、外部境界を示す GM\_Ring と、この GM\_SurfaceBoundary が関連付く。
2. 役割名 interior によって、内部境界を示す GM\_Ring と、この GM\_SurfaceBoundary が関連付く。
3. [上位型から継承する関連]Coordinate Reference System 関連

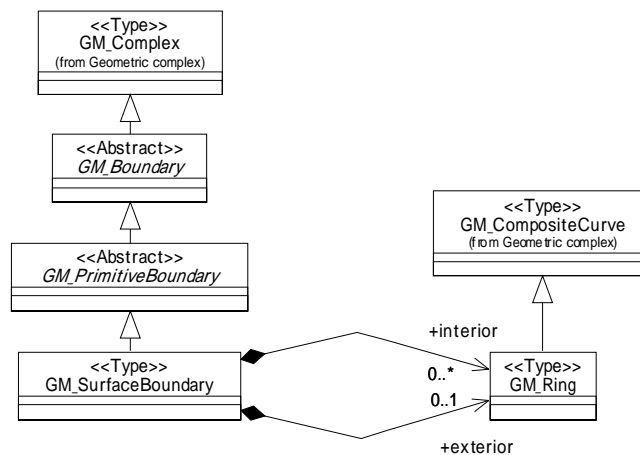


図7-4 GM\_Boundary 及びその下位クラス

### 7.2.2.4 GM\_Ring

幾何輪型である GM\_Ring (図 7-4) は、GM\_SurfaceBoundary を構成するために用いる。GM\_Ring は、列の各 GM\_OrientableCurve の終点が次の GM\_OrientableCurve の始点となる点で、GM\_CompositeCurve と類似した構造をもつ。列は円形なので、この法則は常に成り立つ。各輪はすべての境界と同様に輪体で単純とする。

## 属性

1. [上位型から継承する属性]orientation:Sign

## 関連

1. [上位型から継承する関連]Coordinate Reference System 関連
2. [上位型から継承する関連]Oriented 関連
3. [上位型から継承する関連]Composition 関連

### 7.2.2.5 GM\_Primitive

幾何プリミティブ型である GM\_Primitive (図 7-5) は、幾何プリミティブの抽象基底型である。幾何プリミティブは、他のプリミティブに分解することができない。曲線及び曲面はそれぞれが複数の曲線分及び曲面分で構成されているが、この関係は合成であり、曲線分及び曲面分が他の幾何プリミティブの部分となってはならない。

## 関連

1. Oriented 関連は、一又は二次元の各 GM\_Primitive が役割名 proxy によって、それぞれ異なる方向性の二つの GM\_OrientablePrimitive への関連を集成としてもつことができることを示す。
2. [上位型から継承する関連]Coordinate Reference System 関連

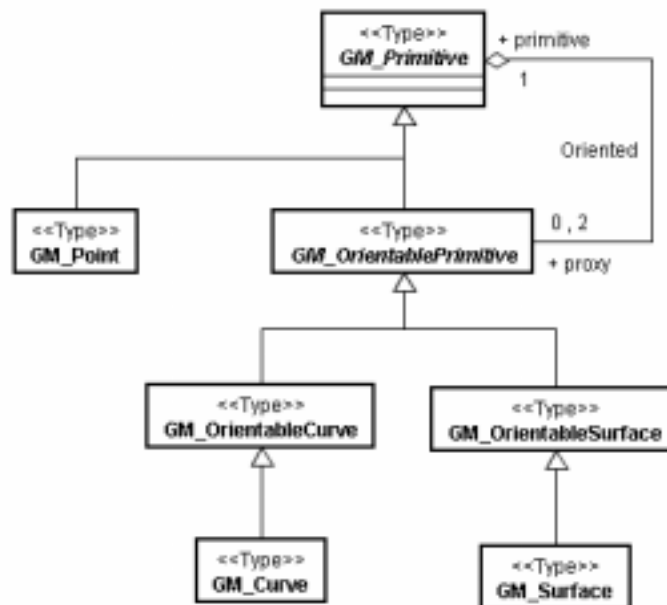


図7-5 GM\_Primitive とその下位クラス

### 7.2.2.6 GM\_Point

点型である GM\_Point (図 7-6) は、ただ一つの点からなる幾何オブジェクトを表す基本的なデータ型とする。

## 属性

1. position:DirectPosition は、GM\_Point がもつ位置となる座標値を表す。

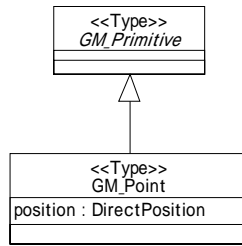


図7-6 GM\_Point

## 関連

1. [上位型から継承する関連]Coordinate Reference System 関連

### 7.2.2.7 GM\_OrientablePrimitive

有向幾何プリミティブ型である GM\_OrientablePrimitive (図 7-7) は、ある幾何プリミティブに対して反対方向の新しい幾何オブジェクトを作成できるプリミティブである。曲線の場合、その向きは曲線をたどる方向を反映する。曲面の境界として用いる場合、囲まれる曲面は、有向曲線の“左”にあるものとする。

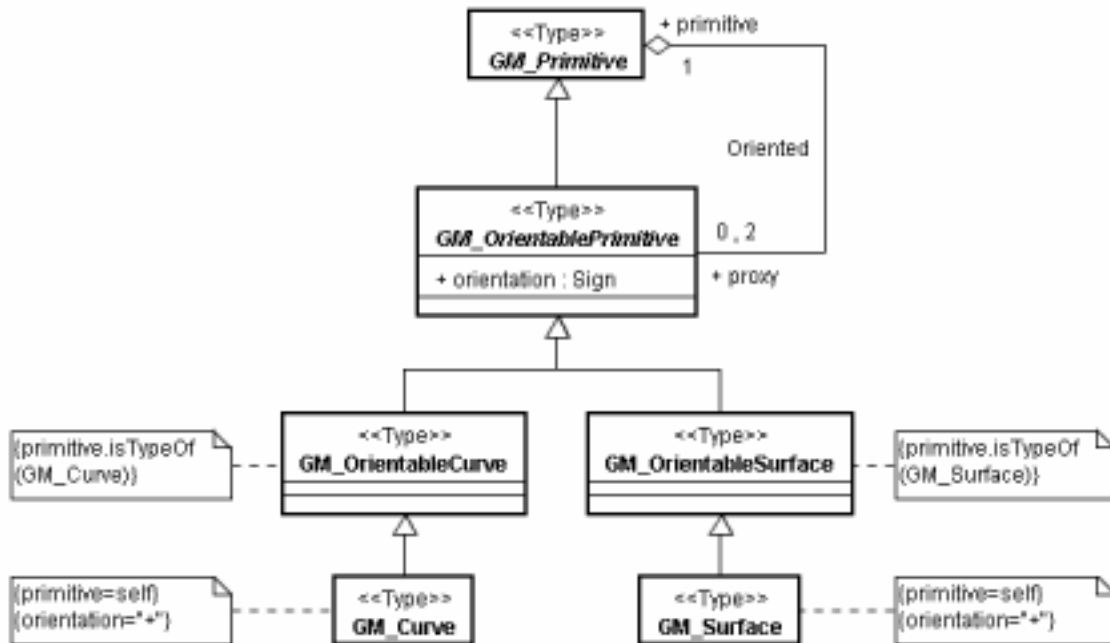


図7-7 GM\_OrientablePrimitive とその下位クラス

## 属性

1. orientation:Sign は、GM\_OrientablePrimitive が表現する方向が正又は負のいずれかを示す。

## 関連

1. Oriented 関連は、役割名 primitive によって GM\_Primitive と、この GM\_OrientablePrimitive を関連付ける。方向性 (orientation) が “+” (正) の場合、GM\_OrientablePrimitive は対応する GM\_Curve 又は GM\_Surface でなければならない。
2. [上位型から継承する関連]Coordinate Reference System 関連

### 7.2.2.8 GM\_OrientableCurve

有向曲線型である GM\_OrientableCurve (図 7-7) は、二次元の幾何オブジェクトである GM\_Surface を構成する際及び GM\_CompositeCurve の構成要素となる際に用いる。

#### 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_OrientableCurve が表現する方向が正又は負のいずれかを示す。

#### 関連

1. [上位型から継承する関連]Oriented 関連は、GM\_Primitive (端点 primitive の先につながる) と、この GM\_OrientableCurve を関連付ける。方向性が“+”の場合、GM\_OrientableCurve は GM\_Curve でなければならない。方向性が“-”の場合、GM\_OrientableCurve は、曲線上の移動の意味で逆向きにパラメタ化するような、他の GM\_Curve と関連付く。
2. [上位型から継承する関連]Coordinate Reference System 関連

#### 制約

1. GM\_OrientableCurve が primitive によって関連付く先の型は GM\_Curve でなくてはならない。

### 7.2.2.9 GM\_OrientableSurface

有向曲面型である GM\_OrientableSurface (図 7-7) は、GM\_Object から継承する Coordinate Reference System 関連、GM\_OrientablePrimitive を継承する orientation 属性及び Oriented 関連をもつ曲面である。GM\_OrientableSurface は立体である GM\_Solid を構成するために必要となる。このプロファイルでは立体をプロファイルの対象としていないが、曲面である GM\_Surface を矛盾なく定義するため及び必要に応じて拡張を可能とするために GM\_Surface の上位クラスとして記載している。

### 7.2.2.10 GM\_Curve

曲線型である GM\_Curve (図 7-8) は、一次元幾何の基本的な要素とする。

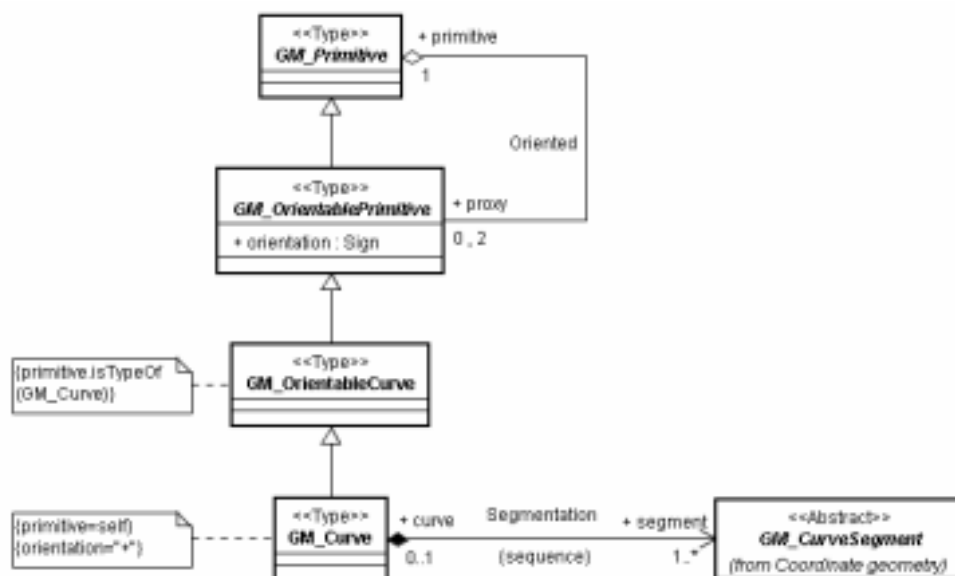


図7-8 GM\_Curve



## 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_Curve が表現する方向を示し、必ず“+”（正）となる。

## 関連

1. Segmentation 関連は、GM\_CurveSegment（端点 segment の先につながる）と、この GM\_Curve を関連付ける合成である。それぞれが曲線の一部に沿った点列の直接位置を定義するような、GM\_Curve の一連の構成要素を示す。GM\_CurveSegment の順番は GM\_Curve を描くのに用いる順番とする。
2. [上位型から継承する関連]Oriented 関連は、役割名 primitive によって自分自身と関連付け、役割名 proxy によって、異なる方向の二つの GM\_OrientableCurve と関連付ける。
3. [上位型から継承する関連]Coordinate Reference System 関連

## 制約

1. GM\_Curve の方向性は必ず“+”であり、Oriented 関連の役割名 primitive によって関連付く先は、自分自身とする。

### 7.2.2.11 GM\_Surface

曲面型である GM\_Surface（図 7-9）は二次元幾何の基本的な要素である。

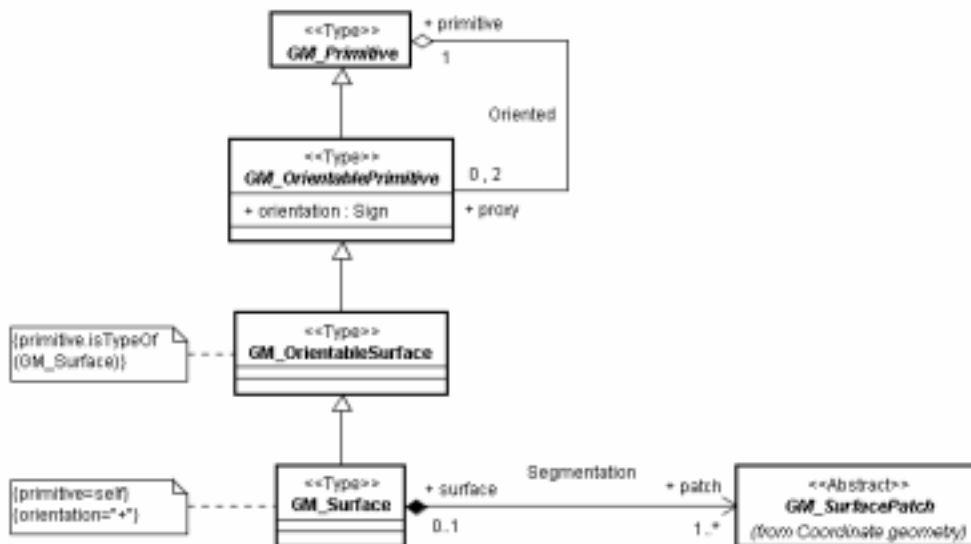


図7-9 GM\_Surface

## 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_Surface が表現する方向を示し、必ず“+”（正）となる。

## 関連

1. Segmentation 関連は、GM\_SurfacePatch（端点 patch の先につながる）と、この GM\_Surface を関連付ける合成である。曲面を形成する互いに隣接する GM\_SurfacePatch の集合と関係させる。
2. [上位型から継承する関連]Oriented 関連は、役割名 primitive によって自分自身と関連付け、役割名 proxy によって、異なる方向の二つの GM\_OrientableSurface と関連付ける。
3. [上位型から継承する関連]Coordinate Reference System 関連

## 制約

1. GM\_Surface の方向性は必ず“+”（正）であり、Oriented 関連の役割名 primitive によって

関連付く先は、自分自身とする。

### 7.2.3 Coordinate geometry パッケージ

Coordinate geometry (座標幾何) パッケージは、座標値を記述するために必要となる型を定義する。

#### 7.2.3.1 DirectPosition

直接位置データ型である DirectPosition (図 7-10) は、ある座標参照系上の位置を保持する。座標参照系との関係は7.5で述べる。

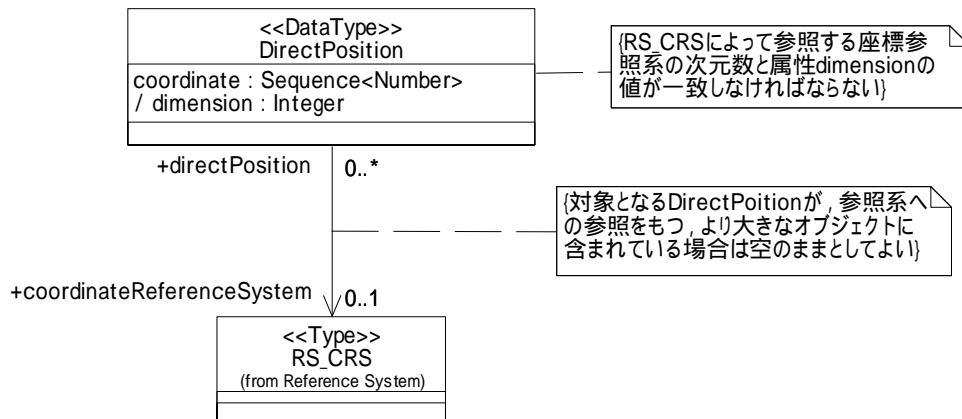


図7-10 DirectPosition

#### 属性

1. coordinate:Sequence<Number>は、指定した参照系におけるこの位置の座標を保持する数値 (Number) の列とする。
2. dimension:Integer は座標列の長さ (項数) とする。これは参照系によって決まり、省略することが可能である。

#### 関連

1. 役割名 coordinateReferenceSystem によって、RS\_CRS と、この DirectPosition は任意に関連付く。対象となる DirectPosition が参照系への参照をもつ、より大きなオブジェクトに含まれている場合は空のままとしてよい。

#### 制約

1. 属性 dimension の値と、属性 coordinate の項数、及び参照系が指定した次元数は一致しなくてはならない。

#### 7.2.3.2 GM\_PointRef

点参照型である GM\_PointRef (図 7-11) は、存在する点を参照するために用いる。

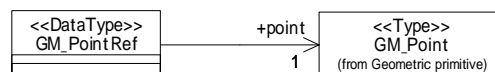


図7-11 GM\_PointRef

#### 関連

1. 役割名 point によって、この GM\_PointRef は、GM\_Point と関連付く。

### 7.2.3.3 GM\_Position

位置データ型である GM\_Position (図 7-12) は、座標として直接 (direct) 又は GM\_Point への参照として間接 (indirect) のどちらかで位置を識別するために用いる共用体である。

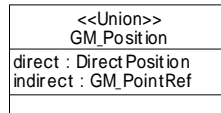


図7-12 GM\_Position

**属性** 以下の属性のうち、いずれかが存在する。

1. direct:DirectPosition は、DirectPosition を型にもつことにより、座標によって位置を識別する。
2. indirect:GM\_PointRef は、GM\_Point への参照をもつ GM\_PointRef を型にもつことにより、間接的に位置を識別する。

### 7.2.3.4 GM\_PointArray

点配列データ型である GM\_PointArray (図 7-13) は、点の列を示すデータ型である。

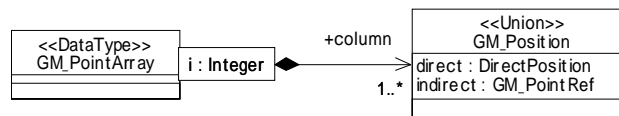


図7-13 GM\_PointArray

#### 関連

1. 役割名 column によって、この GM\_PointArray は一つ以上の GM\_Position と関連付く。

### 7.2.3.5 GM\_CurveSegment

曲線分抽象型である GM\_CurveSegment (図 7-14) は、GM\_Curve の中の同じ内挿法で表現する部分を定義する。GM\_CurveSegment は最大でも一つの GM\_Curve に属する。

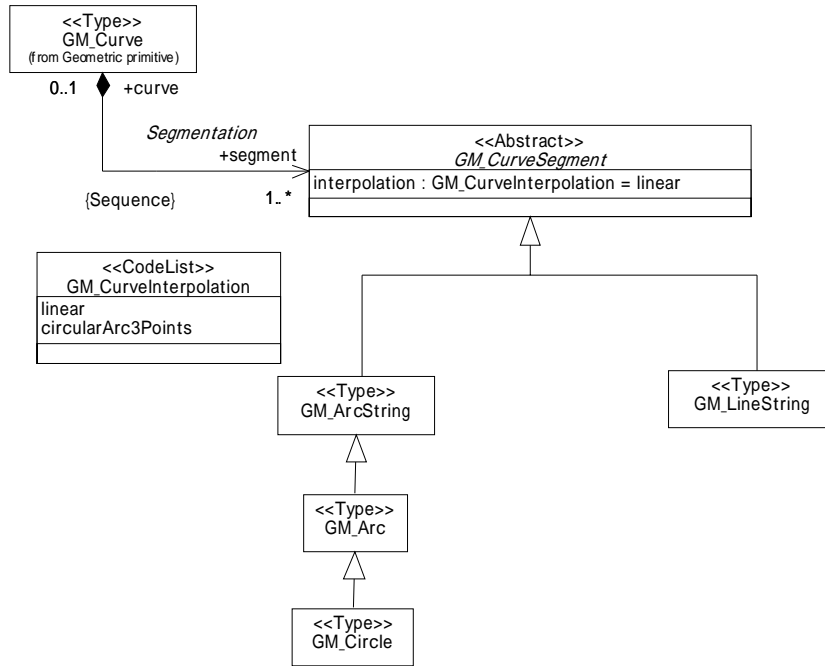


図7-14 GM\_CurveSegment

### 属性

1. interpolation:GM\_CurveInterpolation は、この曲線分に用いる曲線の内挿法を指定する。“linear”（線形）とは、この内挿法が controlPoint の連続する各対の間を結ぶ直線上の DirectPosition を返すことを示す。“circularArc3Points”（三点円弧）は、この内挿法が、controlPoint の連続した三点の組ごとに、二点目の制御点がこの制御点列の始めからの偶数番目となり、一点目から二点目を通り三点目に至る円弧上の DirectPosition を返す。制御点列は奇数の要素をもたなければならない。

### 7.2.3.6 GM\_LineString

折れ線型である GM\_LineString（図 7-15）は、GM\_CurveSegment を継承した折れ線を表す型である。

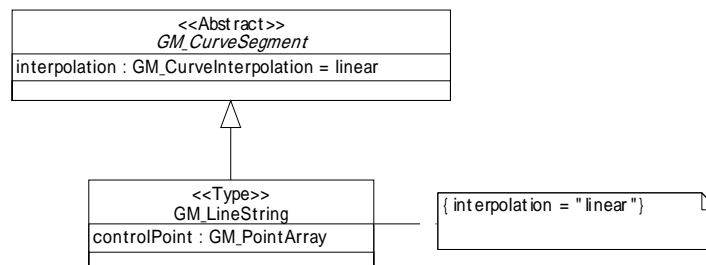


図7-15 GM\_LineString

### 属性

1. [上位型から継承する属性] interpolation:GM\_CurveInterpolation は、この折れ線分に用いる内挿法を指定するため、値は“linear”となり、controlPoint で与えられた点を直線的

に内挿することを示す。

- controlPoint : GM\_PointArray は、この折れ線分を構成する位置の列であり、この列の一点目は GM\_LineString の始点であり、列の最後の点は終点となる。

### 7.2.3.7 GM\_ArcString

弧列型である GM\_ArcString (図 7-16) は、GM\_CurveSegment を継承した円弧を表す型である。

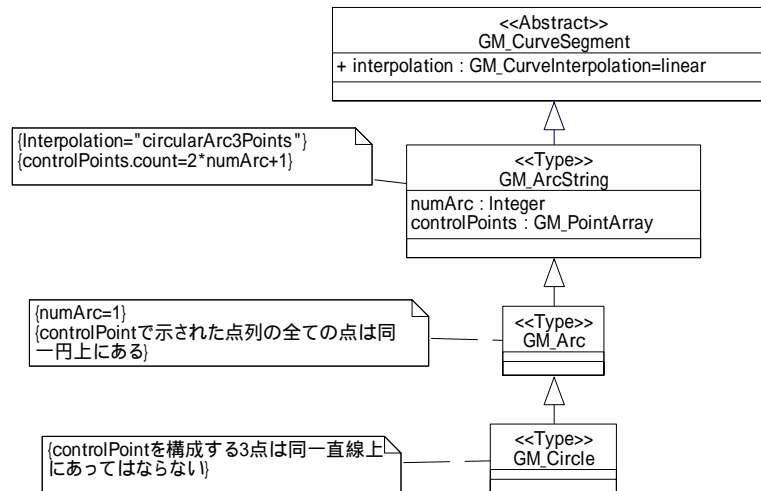


図7-16 GM\_ArcString とその下位クラス

#### 属性

- [上位型から継承する属性] interpolation:GM\_CurveInterpolation は、この弧列に用いる内挿法を指定するため、値は“circularArc3Points”となり、controlPoint で与えられた点列の奇数番目から始まる三つの連続した点によって円弧が決定される。
- numArc:Integer は、この GM\_ArcString における円弧の数とする。
- controlPoints : GM\_PointArray は、この弧列を構成する点の列である。

#### 制約

- 属性 interpolation の値は必ず“circularArc3Points”でなくてはならない。
- 属性 controlPoints を構成する点列の数は、属性 numArc で与えられた整数値を倍にした値に 1 を加算した値と一致しなくてはならない。

### 7.2.3.8 GM\_Arc

弧型である GM\_Arc (図 7-16) は、GM\_ArcString を継承した円弧を表す型である。GM\_Arc は三点で定義され、その三点によって決まる円の一点目から始まり、二点目を通して三点目に至る弧からなる。この三点が同一直線上にあってはならない。

#### 属性

- [上位型から継承する属性] interpolation:GM\_CurveInterpolation は、この円弧に用いる内挿法を指定するため、値は“circularArc3Points”となる。
- [上位型から継承する属性] numArc:Integer は、この GM\_Arc における円弧の数であり、値は“1”となる
- [上位型から継承する属性] controlPoints : GM\_PointArray は、この円弧を構成する三点からなる点の列である。

#### 制約

- 属性 controlPoints を構成する三点は、全て同一円上になくなくてはならない。

### 7.2.3.9 GM\_Circle

円型である GM\_Circle (図 7-16) は, GM\_Arc を継承した円を表す型である。

#### 属性

1. [上位型から継承する属性] interpolation:GM\_CurveInterpolation は, この円に用いる内挿法を指定するため, 値は “circularArc3Points” となる。
2. [上位型から継承する属性] numArc:Integer は, この GM\_Circle における円弧の数であり, 値は “1” となる。
3. [上位型から継承する属性] controlPoints : GM\_PointArray は, この円を構成する三点からなる点の列である。

**備考** GM\_Circle を構成する点は四点必要であり, 始点と終点は一致する。

#### 制約

1. 属性 controlPoints を構成する三点は, 同一直線上にあってはならない。

### 7.2.3.10 GM\_SurfacePatch

曲面分抽象型である GM\_SurfacePatch (図 7-17) は, GM\_Surface の中の同じ内挿法で表現する部分を定義する。GM\_SurfacePatch は最大でも一つの GM\_Surface に属する。

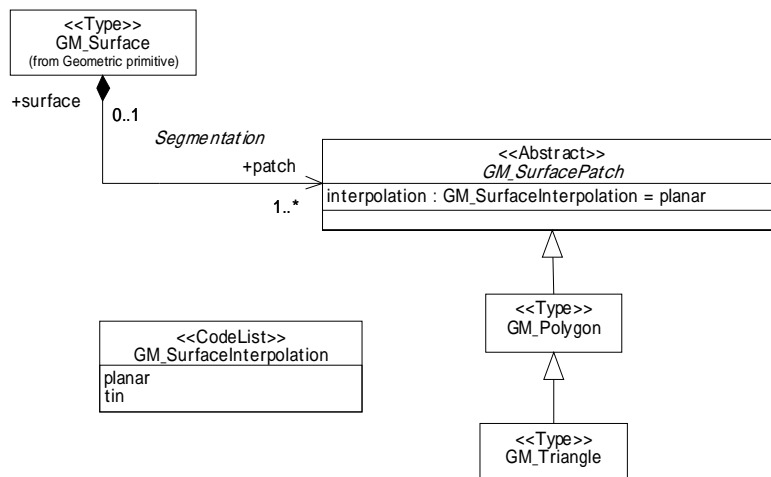


図7-17 GM\_SurfacePatch

#### 属性

1. interpolation:GM\_SurfaceInterpolation は, この GM\_SurfacePatch に用いる内挿法を指定する。  
“planar” (平面) は, この内挿法が, 単一の平面上の点を返し, 境界はこの平面上に含まれることを示す。“tin” (不規則三角網) は, 制御点は隣接する複数の三角形を構成し, それぞれが小平面分を形成することを示す。

### 7.2.3.11 GM\_PolyhedralSurface

多面体面型である GM\_PolyhedralSurface (図 7-18) は, 共通な境界曲線に沿って結合した GM\_Polygon から構成する GM\_Surface とする。

## 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_PolyhedralSurface が表現する方向を示し、必ず正となる。

## 関連

1. [上位型から継承する関連]Segmentation 関連は、GM\_Polygon（端点 patch の先につながる）と、この GM\_PolyhedralSurface を関連付ける合成である。GM\_PolyhedralSurface を形成する互いに隣接する GM\_Polygon の集合と関係させる。
2. [上位型から継承する関連]Oriented 関連は、役割名 primitive によって自分自身と関連付け、役割名 proxy によって、異なる方向の二つの GM\_OrientableSurface と関連付ける。
3. [上位型から継承する関連]Coordinate Reference System 関連

## 制約

1. GM\_PolyhedralSurface の方向性は必ず “+” であり、Oriented 関連によって関連付く先は、自分自身とする。

### 7.2.3.12 GM\_TriangulatedSurface

三角網曲面型である GM\_TriangulatedSurface（図 7-18）は、複数の GM\_Triangle だけから構成した GM\_PolyhedralSurface とする。

## 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_TriangulatedSurface が表現する方向を示し、必ず正となる。

## 関連

1. [上位型から継承する関連]Segmentation 関連は、GM\_Triangle（端点 patch の先につながる）と、この GM\_TriangulatedSurface を関連付ける合成である。GM\_TriangulatedSurface を形成する互いにつながった GM\_Triangle の集合と関係させる。
2. [上位型から継承する関連]Oriented 関連は、役割名 primitive によって自分自身と関連付け、役割名 proxy によって、異なる方向の二つの GM\_OrientableSurface と関連付ける。
3. [上位型から継承する関連]Coordinate Reference System 関連

## 制約

1. GM\_TriangulatedSurface の方向性は必ず “+” であり、Oriented 関連によって関連付く先は、自分自身とする。

### 7.2.3.13 GM\_Tin

不規則三角網曲面型である GM\_Tin（図 7-18）は、ドローネ・アルゴリズム又はこれに抑止線、傾斜変換線及び三角形の最大辺長に対する考慮を補った同様のアルゴリズムを使用した GM\_TriangulatedSurface とする。

## 属性

1. [上位型から継承する属性]orientation:Sign は、GM\_Tin が表現する方向を示し、必ず正となる。
2. stopLines:Set<GM\_LineString>は、曲面の連続性又は正規性の局所的に問題がある部分の線（抑止線）である。これらの異常な領域では抑止線と交わる三角形は不規則三角網曲面から取り除き、曲面の穴として残すものとする。
3. breakLines:Set<GM\_LineString>は、曲面の局所的な稜線又は（流水線のような）谷線を表す曲面の形の決定的な本質となる線（傾斜変換線）とする。
4. maxLength:Distance は、三角形の最大辺長を示し、これを越える三角形の辺については、その辺に隣接する三角形を曲面から削除する。これによりデータが適切な計算の保証に十分な密度をもたない曲面の領域を取り除くことができる。

5. controlPoint[3..n]:GM\_Position は、不規則三角網の三角形の頂点（制御点）の集合である。

#### 関連

1. [上位型から継承する関連]Segmentation 関連は、GM\_Triangle（端点 patch の先につながる）と、この GM\_Tin を関連付ける合成である。GM\_Tin を形成する互いにつながった GM\_Triangle の集合と関係させる。
2. [上位型から継承する関連]Oriented 関連は、役割名 primitive によって自分自身と関連付け、役割名 proxy によって、異なる方向の二つの GM\_OrientableSurface と関連付ける。
3. [上位型から継承する関連]Coordinate Reference System 関連

#### 制約

1. GM\_Tin の方向性は必ず “+” であり、Oriented 関連によって関連付く先は、自分自身とする。

#### 7.2.3.14 GM\_Polygon

ポリゴン型である GM\_Polygon（図 7-18）は、境界となる曲線の集合によって表現された曲面である。境界となる曲線は同一平面上に存在し、GM\_Polygon は平面を示す。

#### 属性

1. [上位型から継承する属性]interpolation:GM\_SurfaceInterpolation は、この GM\_Polygon に用いる内挿法を “planar” と指定する。
2. boundary:GM\_SurfaceBoundary は、この GM\_Polygon の境界を表す。GM\_SurfaceBoundary の属性である exterior はこの GM\_Polygon の外部境界となり、interior は内部境界となる。

#### 7.2.3.15 GM\_Triangle

三角形型である GM\_Triangle（図 7-18）は、三つの頂点で定義した平面上の GM\_Polygon とする。

#### 属性

1. [上位型から継承する属性]interpolation:GM\_SurfaceInterpolation は、この GM\_Triangle に用いる内挿法を指定する。GM\_Tin の構成要素となる場合は、値 “tin” をとる。
2. [上位型から継承する属性]boundary:GM\_SurfaceBoundary は、この GM\_Triangle の境界を表す。GM\_SurfaceBoundary の属性である exterior はこの GM\_Triangle の外周を示す。interior は存在しない。
3. corners[3]:GM\_Position は、GM\_Triangle の頂点となる点である。



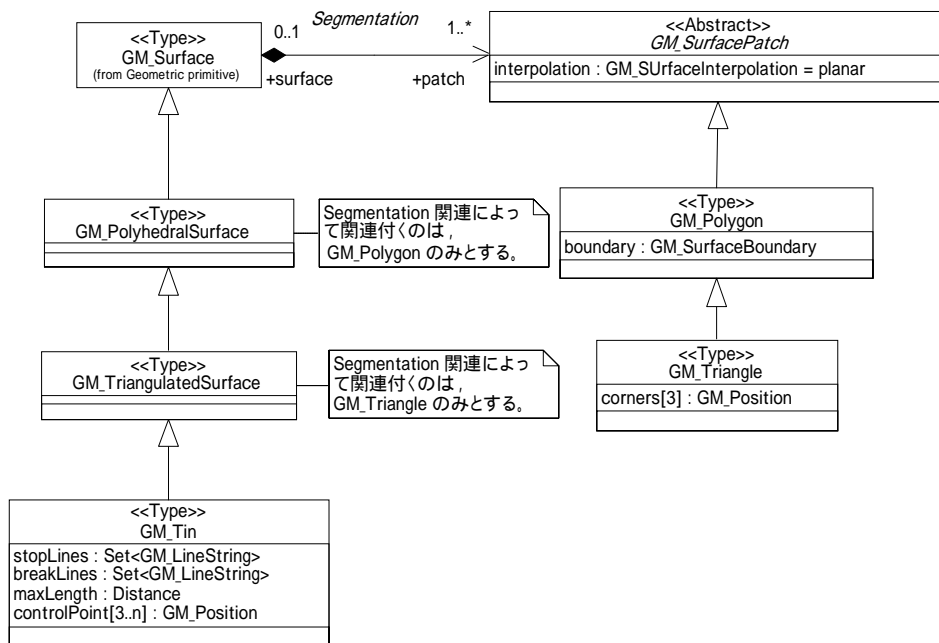


図7-18 GM\_Surface 及び GM\_SurfacePatch の下位クラス

## 7.2.4 Geometric complex パッケージ

Geometric complex (幾何複体) パッケージは、共通の座標系において、その内部に交差のない幾何プリミティブの集合とする。

### 7.2.4.1 GM\_Complex

幾何複体型である GM\_Complex (図 7-19) は、幾何的に交差のない単純な GM\_Primitive の集合とする。GM\_Point 以外の GM\_Primitive が、ある GM\_Complex に含まれている場合、その GM\_Complex の中に、その GM\_Primitive の境界を形成するより低い次元の GM\_Primitive が存在する。

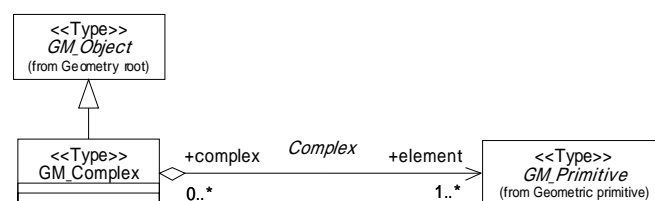


図7-19 GM\_Complex

### 関連

1. Complex 関連は、GM\_Complex に含まれる一つ以上の GM\_Primitive (端点 element の先につながる) と関連付く。
2. [上位型から継承する関連]Coordinate Reference System 関連

### 7.2.4.2 GM\_CompositeCurve

合成曲線型である GM\_CompositeCurve (図 7-20) は, GM\_OrientableCurve の集合となる。GM\_CompositeCurve は GM\_OrientableCurve を継承するため, (最初の曲線を除く) 各曲線が直前の曲線の終点から始まるという方法で方向が揃った有向曲線となる。



図7-20 GM\_CompositeCurve

## 属性

1. [上位型から継承する属性]orientation:Sign は, GM\_CompositeCurve が表現する方向が正又は負のいずれかを示す。

## 関連

1. [上位型から継承する関連]Oriented 関連は, 役割名 proxy によって, それぞれ異なる方向性の二つの GM\_OrientablePrimitive への関連をもつことを示し, 役割名 primitive によって曲線上の移動の意味で逆向きにパラメタ化するような GM\_CompositeCurve と, この GM\_CompositeCurve を関連付ける。つまり方向性が “+” の場合, 同じ場所にあり, 方向性が “-” となる GM\_CompositeCurve と関連付く。他方, 方向性が “-” の場合は, 同じ場所にあり, 方向性が “+” となる GM\_CompositeCurve と関連付く。

**備考** 方向性が “-” となる GM\_CompositeCurve はそれ自身が GM\_Surface の境界曲線となるとき又はより大きな GM\_CompositeCurve の構成要素となるときのみ用いられる。

2. Composition 関連は, GM\_CompositeCurve に含まれる一つ以上の GM\_OrientableCurve (端点 generator の先につながる) と関連付く。
3. [上位型から継承する関連]Coordinate Reference System 関連

## 7.2.5 Geometric aggregates パッケージ

Geometric aggregates (幾何集成) パッケージは, 共通の座標系において, ある特定の型を集めることのできる型を定義する。

### 7.2.5.1 GM\_Aggregate

幾何集成型である GM\_Aggregate (図 7-21) は, GM\_Object の集まりによって構成される。

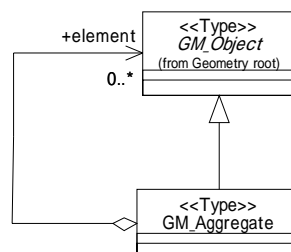


図7-21 GM\_Aggregate

## 関連

1. GM\_Aggregate は, 役割名 element によって自身を構成する GM\_Object と関連付く。
2. [上位型から継承する関連]Coordinate Reference System 関連

## 7.3 位相スキーマ

### 7.3.1 一般

位相属性の最も生産的な使用法は、計算幾何処理の高速化である。

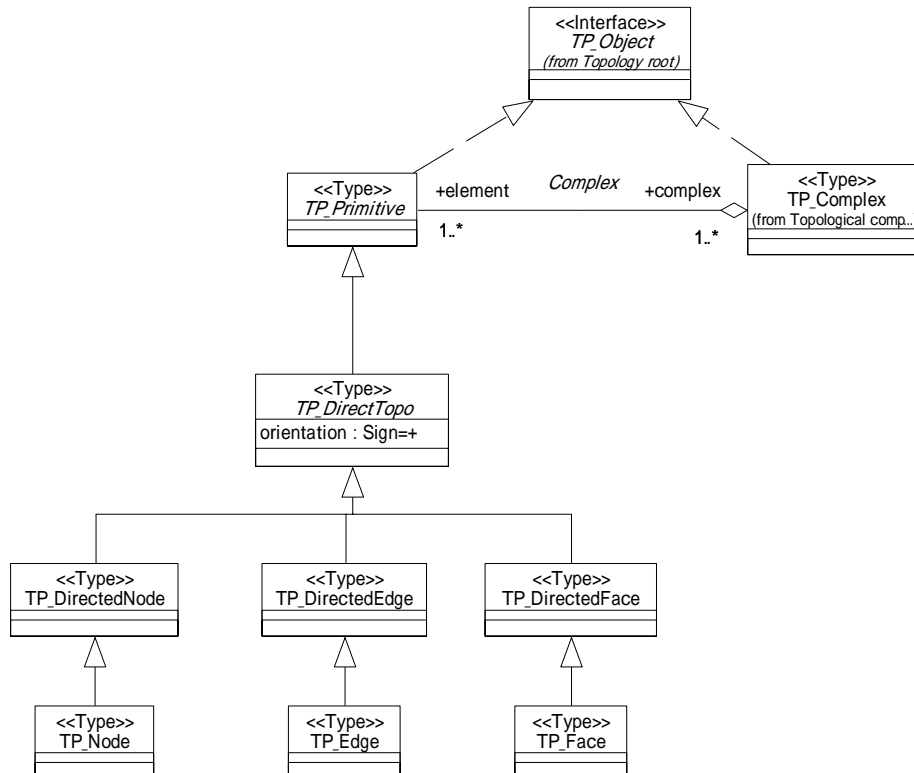


図7-22 位相スキーマ

### 7.3.2 Topology root パッケージ

#### 7.3.2.1 TP\_Object

TP\_Object は、位相複体及び位相プリミティブの基底型となる抽象クラスである。

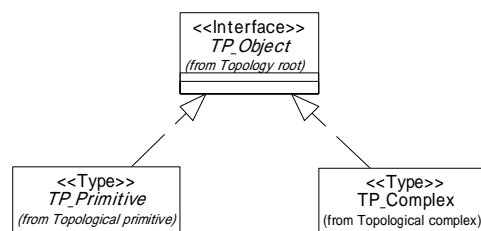


図7-23 TP\_Object

### 7.3.3 Topological primitive パッケージ

Topological primitive パッケージは各次元の全ての位相プリミティブを含み、その構造上の関係を表現するためのクラスに対応する。このプロファイルでは二次元までの位相プリミティブを定義する。

### 7.3.3.1 TP\_Primitive

位相プリミティブ型 (TP\_Primitive) は、位相プリミティブの抽象基底クラスであり、位相複体を構成する要素となる。位相プリミティブは、システム内でさらに他のプリミティブに分解することができない。

#### 関連

1. Center 関連は、この TP\_Primitive を参照する向きをもつ位相オブジェクト（端点 proxy の先につながる）を識別する。

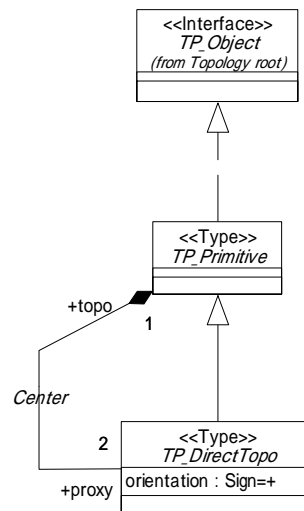


図7-24 TP\_Primitive

### 7.3.3.2 TP\_DirectedTopo

有向位相型 (TP\_DirectedTopo) は、向きをもつ位相複体及び位相プリミティブの基底型となる抽象クラスである。

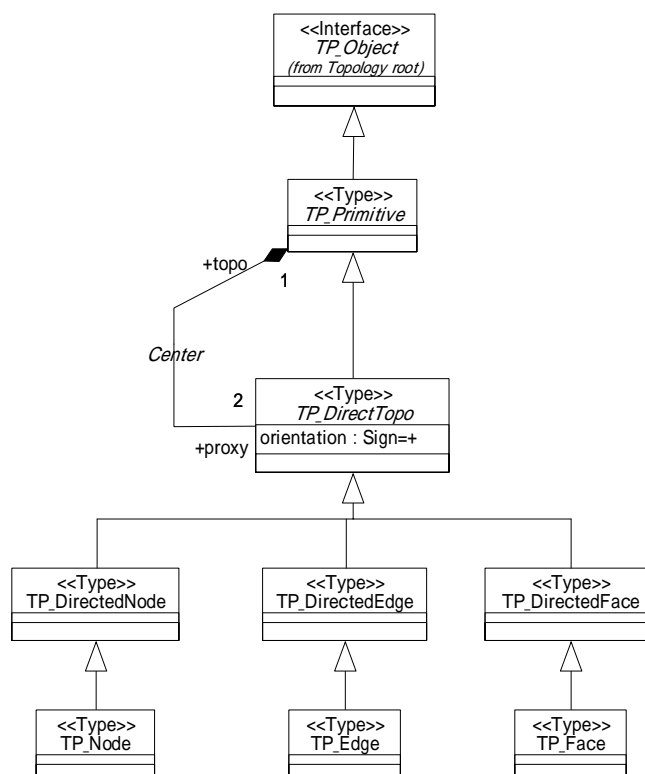


図7-25 位相プリミティブの下位クラス

### 属性

1. orientation:Sign は、この TP\_DirectedTopo がその元になる TP\_Primitive に対する向きとする。

### 関連

1. Center 関連は、この TP\_DirectedTopo の元となる TP\_Primitive (端点 topo の先につながる) を識別する。

#### 7.3.3.3 TP\_DirectedNode

TP\_DirectedNode (有向ノード型) は、有向位相オブジェクトである。

### 属性

1. [上位型から継承する属性]orientation:Sign は、この TP\_DirectedNode がその元になる TP\_Node に対する向きとする。TP\_Edge の始点となる場合は負の TP\_DirectedNode となり、TP\_Edge の終点となる場合は正の TP\_DirectedNode となる。

### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedTopo の元となる TP\_Primitive (端点 topo の先につながる) を識別する。

#### 7.3.3.4 TP\_Node

TP\_Node (ノード型) は零次元の位相プリミティブである。

### 属性

1. [上位型から継承する属性]orientation:Sign は、TP\_Node の方向性を表し、常に正とする。

### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedNode の元となる TP\_Node (端点 topo の先につながる) を識別するが、ここでは自分自身を参照する。
2. CoBoundary 関連 (図 7-26参照) は、この TP\_Node に入るエッジ (正の TP\_DirectedEdge) 及び出るエッジ (負の TP\_DirectedEdge) を示す、TP\_DirectedEdge (端点 spoke につながる) と関連付ける。

#### 7.3.3.5 TP\_DirectedEdge

TP\_DirectedEdge (有向エッジ型) は、有向位相オブジェクトである。元となる TP\_Edge が必要なときにその代替として機能する。

##### 属性

1. [上位型から継承する属性]orientation:Sign は、この TP\_DirectedEdge がその元になる TP\_Edge に対する向きとする。

##### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedEdge の元となる TP\_Edge (端点 topo の先につながる) を識別する。

#### 7.3.3.6 TP\_Edge

TP\_Edge (エッジ型) 位相のための一次元プリミティブである。TP\_Edge は正の TP\_DirectedEdge と等価である。

##### 属性

1. [上位型から継承する属性]orientation:Sign は、TP\_Edge の方向性を表し、常に正とする。

##### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedEdge の元となる TP\_Edge (端点 topo の先につながる) を識別するが、ここでは自分自身を参照する。
2. Boundary 関連 (図 7-26参照) は、この TP\_Edge の始点 (負の TP\_DirectedNode) 及び終点 (正の TP\_DirectedNode) を示す、TP\_DirectedNode (端点 boundary につながる) と関連付ける。

#### 7.3.3.7 TP\_DirectedFace

TP\_DirectedFace (有向フェイス型) は、有向位相オブジェクトである。元となる TP\_Face が必要なときにその代替として機能する。

##### 属性

1. [上位型から継承する属性]orientation:Sign は、この TP\_DirectedFace がその元になる TP\_Face に対する向きとする。

##### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedFace の元となる TP\_Face (端点 topo の先につながる) を識別する。

#### 7.3.3.8 TP\_Face

TP\_Face (フェイス型) 位相のための二次元プリミティブである。TP\_Face は正の TP\_DirectedFace と等価である。

##### 属性

1. [上位型から継承する属性]orientation:Sign は、TP\_Face の方向性を表し、常に正とする。

##### 関連

1. [上位型から継承する関連]Center 関連は、TP\_DirectedFace の元となる TP\_Face (端点 topo

の先につながる)を識別するが、ここでは自分自身を参照する。

- Boundary 関連は、この TP\_Face の始点 (正の TP\_DirectedEdge) 及び終点 (負の TP\_DirectedEdge) を示す、TP\_DirectedEdge (端点 boundary につながる) と関連付ける。

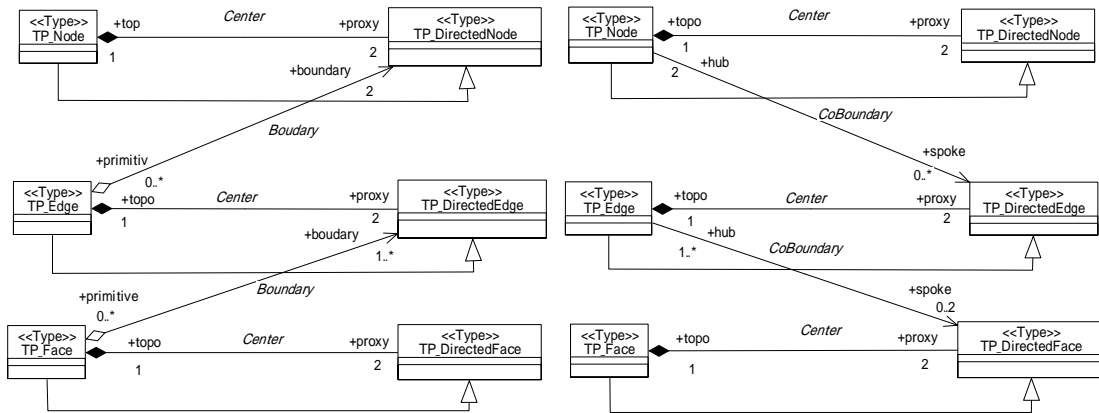


図7-26 位相クラスの関連

### 7.3.4 Topological complex パッケージ

Topological complex パッケージは、TP\_Complex の生成を行う追加のクラスを提供する。

#### 7.3.4.1 TP\_Complex

TP\_Complex (位相複体型) は、位相プリミティブの集合である。

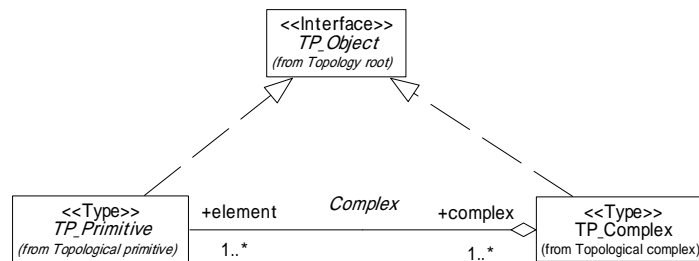


図7-27 TP\_Complex

#### 関連

- Complex 関連は、この TP\_Complex に、自身を構成する TP\_Primitive (端点 element につながる) と関連付ける。

### 7.4 幾何実現

#### 7.4.1 TP\_Primitive の幾何実現

位相プリミティブは幾何複体の要素である単一次元の幾何プリミティブと対応する。

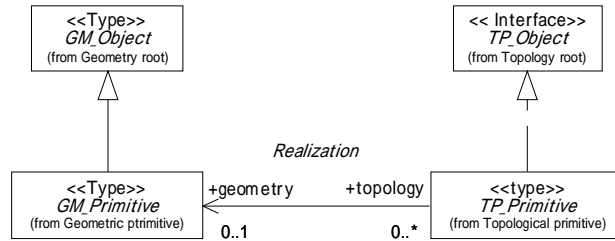


図7-28 位相プリミティブの幾何実現

### 関連

1. Realization 関連は、この TP\_Primitive をこれと対応する GM\_Primitive（端点 geometry とつながる）が存在すれば、関連付ける。

### 7.4.2 TP\_Complex の幾何実現

位相複体は、幾何複体から生成する（図 7-29参照）。幾何プリミティブが交差しない幾何複体のみがエラーのない位相複体を生成する。

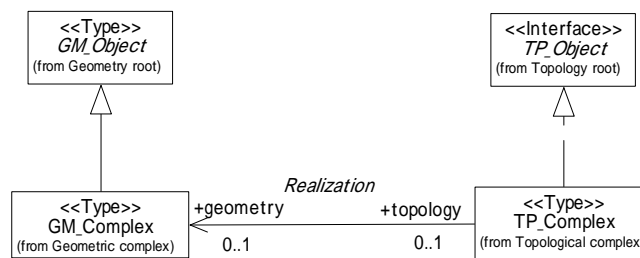


図7-29 位相複体の幾何実現

### 関連

1. Realization 関連は、この TP\_Complex をこれと対応する GM\_Complex（端点 geometry とつながる）が存在すれば、関連付ける。

### 7.5 座標参照系との関連

#### 7.5.1 一般

オブジェクトの幾何の記述に用いられる数学関数は、座標位置の定義に用いる座標参照系の型に依存している。このプロファイルでは、座標参照系は附属書 2 で規定する。

#### 7.5.2 直接位置と座標参照

DirectPosition はある座標参照系上の位置の座標を保持する直接位置データ型であり、役割名 coordinateReferenceSystem によって自身の定義に用いる空間参照系を参照する。



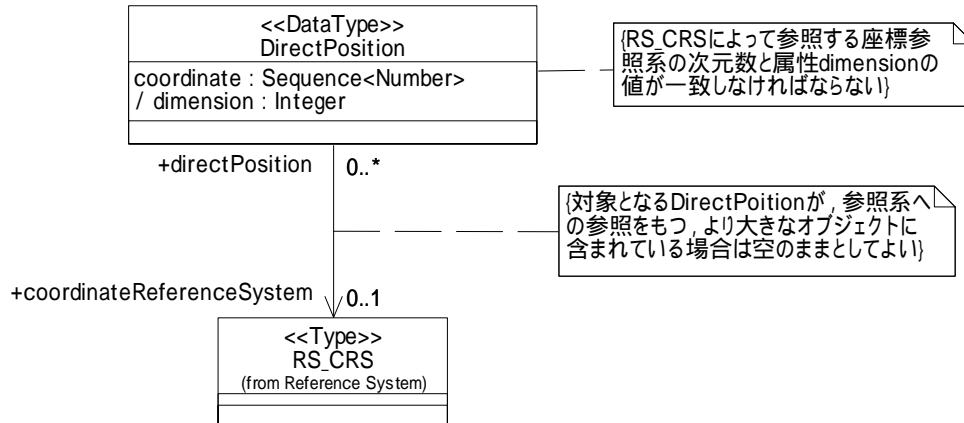


図7-30 直接位置と空間参照

### 関連

- coordinateReferenceSystem によってこの DirectPosition の座標を定めている座標参照系 RS\_CRS と関連付く。ただし、対象となる DirectPosition が参照系への参照をもつより大きなオブジェクトに含まれている場合は空のままとしてよい。この場合は、DirectPosition は暗黙でそれを含むオブジェクトの座標参照系の値をとるものとして扱う。

### 7.5.3 幾何オブジェクトと座標参照

GM\_Object はすべての幾何オブジェクトの基底となる抽象型である。幾何オブジェクトは座標幾何と座標参照系の組み合わせであり、CoordinateReferenceSystem 関連によって自身の定義に用いる座標参照系を参照する。

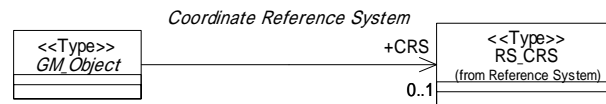


図7-31 幾何オブジェクトと空間参照

### 関連

- CoordinateReferenceSystem 関連は、この GM\_Object の座標幾何の記述に用いる RS\_CRS（端点 CRS とつながる）と関連付ける。この関連が空の場合、GM\_Object は、それを包含する他の GM\_Object の RS\_CRS を用いる。

## 7.6 空間スキーマプロファイル拡張及び制限のための規則

### 7.6.1 一般

空間スキーマプロファイルを拡張及び制限する場合の規則を定める。プロファイルを拡張して用いる場合は、原規格を参照し抽象試験項目群を設定しなければならない。

### 7.6.2 拡張のための規則

このプロファイルで定義したスキーマは、JIS X 7107 で規定する範囲内であれば、以下の規則に則って拡張することができる。

## 規則

1. 空間スキーマ (JIS X 7107) に定義された別の型を用いたい場合は、当該クラスの上位クラスから継承させ、プロファイルに定義する。  
**例** クロソイド曲線分を示す GM\_Clothoid を拡張したい場合は、上位クラスである GM\_CurveSegment から継承し、プロファイルを再定義する。
2. プロファイルに追加したいクラスの上位クラスがプロファイルに存在しない場合は上位クラスも併せてプロファイルに追加する。  
**例** 多曲線型を示す GM\_MultiCurve を拡張したい場合は、その上位クラスとなる GM\_MultiPrimitive がプロファイルに存在しないため、GM\_MultiPrimitive を GM\_Aggregate から継承させ定義し、GM\_MultiCurve を GM\_MultiPrimitive から継承させ、プロファイルに定義する。
3. 追加するクラスが必須の属性及び関連をもつ場合は、もれなく追加する。必須でないものは、目的に照らして判断する。

### 7.6.3 制限のための規則

このプロファイルで定義されたスキーマを制限し、その一部を使用することができる。

## 規則

1. 空間スキーマプロファイルの一部を制限して使用する場合は、プロファイルの修正は行わず、応用スキーマに記載する。  
**例** 円型を示す GM\_Circle を用いないときは、プロファイルを修正し GM\_Circle を削除するのではなく、応用スキーマにおいてその地物の空間属性の定義に、GM\_Circle は用いないことを記述する。

## 8 時間スキーマ

**序文** このプロファイルは、ISO 19108 (Geographic information - Temporal schema) を翻訳し、技術的内容及び規格票の様式を変更することなく作成した JIS X 7108 (地理情報-時間スキーマ) から、時間属性を記述するために必要となる基本的な要素を抽出し、プロファイルとして規定したものである。

### 8.1 スキーマの構造

ここでは、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す。このスキーマは、OMG UML1.4.2 によって記述する。このスキーマは抽象モデルであり、ある実装が JPGIS 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した内容を実現することができなくてはならないが、同じ方法で実装する必要はない。

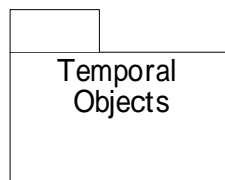


図8-1 時間スキーマの構造

時間スキーマは、時間オブジェクト (Temporal Objects) パッケージからなる (図 8-1参照)。時間オブジェクト (Temporal Objects) パッケージは、時間幾何オブジェクト (8.2参照) 及び時間位相オブジェクト (8.3参照) を定義するが、これらは、地物とデータ集合との時間特性の値として使わなければならない。オブジェクトの時間位置は、附属書 2 で規定する JPGIS 参照系パッケージに記述された参照系と関係付けて指定しなければならない (8.5参照)。

このプロファイルで定義する UML クラス名は、特定の規格を識別するため、二文字の英字及び一文字の下線からなる接頭辞から開始し、時間スキーマで定義するクラスを識別する接頭辞は、TM\_とする。

#### 8.1.1 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、幾何及び位相特性をもつ。時間における点は、時間参照系と関係付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す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持ち、時間における動きは常に前方向ではあるが、時間は二つの方向で示すことができる。

**備考** 時間は、概念レベルでは常に幾何及び位相特性をもつ。しかし、幾何特性又は位相特性のいずれかの一方だけで時間を記述することが可能なこともあり、また、それが望ましいこともある。

時間は、順序尺度及び間隔尺度の二種類の尺度で測定する。順序尺度は時間に関する相対的な位置の情報だけを示すが、間隔尺度は持続時間を測定するための基礎を示す。

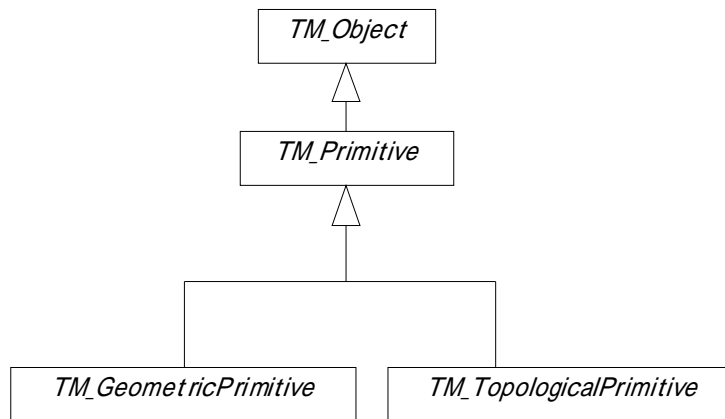


図8-2 時間オブジェクト

### 8.1.2 時間のオブジェクト

地物及びデータ集合の時間特性を示す値として使用する最上位の抽象クラスを `TM_Object` (図8-2参照)とする。時間プリミティブ (`TM_Primitive`) は、不可分な幾何又は位相を表現する抽象クラスとする。`TM_Primitive` は時間幾何を示す `TM_GeometricPrimitive` と時間位相を示す `TM_TopologicalPrimitive` の二つの下位型をもつ。

## 8.2 幾何スキーマ

### 8.2.1 `TM_GeometricPrimitive`

`TM_GeometricPrimitive` は、これ以上不可分な時間幾何を示す抽象クラスである。時間の次元にある二つの幾何プリミティブは、瞬間及び期間とする。`TM_GeometricPrimitive` の下位型 `TM_Instant` は瞬間を、`TM_Period` は期間を表す (図8-3参照)。

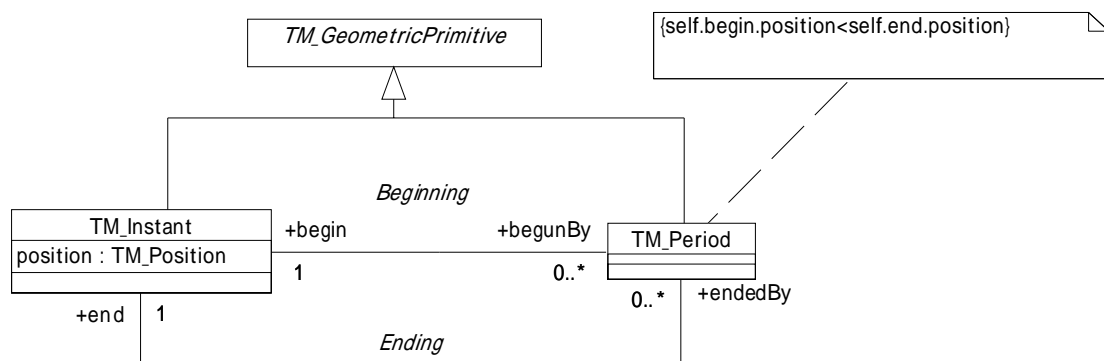


図8-3 時間幾何プリミティブ

### 8.2.2 `TM_Instant`

瞬間 (`TM_Instant`) は、零次元幾何プリミティブであり、時間上の位置を表現する。これは空間における点と同等である。実際には、瞬間は時間間隔であるが、その持続時間は時間尺度の分解能より短い。

## 属性

1. position:TM\_Position は、この TM\_Instant の位置を指定しなければならない。TM\_Position は、8.5で規定する単一の時間参照系に関連付けなければならない。TM\_Instant のインスタンスは他と識別できるオブジェクトとなるが、TM\_Position のインスタンスは、一つのデータ値となる。任意の TM\_Instant の TM\_Position は、異なる時間参照系に関連する TM\_Position によって置換できる。

## 関連

1. Beginning 関連は、期間を示す TM\_Period (端点 begunBy の先につながる) と、この TM\_Instant とを関連付ける。
2. Ending 関連は、期間を示す TM\_Period (端点 endedBy の先につながる) と、この TM\_Instant とを関連付ける。

### 8.2.3 TM\_Period

期間 (TM\_Period) は、時間の範囲を表現する一次元幾何プリミティブとする。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 (瞬間) を境界にもつ開区間をなし、長さ (持続時間) 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し、その長さは二つの時間位置間の時間距離に等しい。

## 関連

1. Beginning 関連は、期間開始位置を示す TM\_Instant (端点 begin の先につながる) と、この TM\_Period を関連付ける。
2. Ending 関連は、期間開始位置を示す TM\_Instant (端点 end の先につながる) と、この TM\_Period を関連付ける。

## 制約

1. {self.begin.position < self.end.position} は、期間開始の時間位置が期間終了の時間位置よりも小さく (より以前に) なければならないことを示す。

## 8.3 位相スキーマ

### 8.3.1 一般

位相は、オブジェクト間の時間連結性に関する情報を示すと共に、時間における地物の順序についての情報を示すこともできる。位相は、時間位置に関する情報は示さない。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、位相関係を明確に記述していることが要件となる場合には、位相を用いてもよい。

### 8.3.2 TM\_TopologicalPrimitive

TM\_TopologicalPrimitive は、これ以上不可分な時間位相を示す抽象クラスを示す。時間情報に該当する二つの位相プリミティブがあるが、その内の一つは零次元のノードであり、他の一つは一次元のエッジである。時間スキーマにおいては、これらは、TM\_TopologicalPrimitive の下位クラスである TM\_Node 及び TM\_Edge で表す (図 8-4参照)。

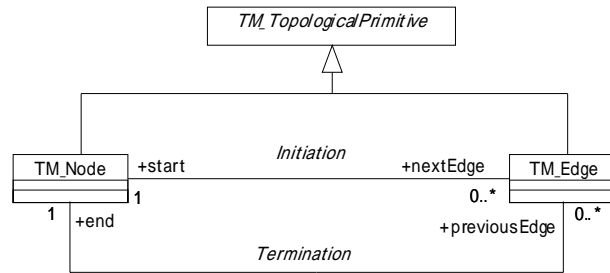


図8-4 時間位相プリミティブ

### 8.3.3 TM\_Node

TM\_Node は時間における零次元位相プリミティブとする。その幾何的実現は TM\_Instant と対応する。

#### 関連

1. Initiation 関連は、この TM\_Node を始点にもつ TM\_Edge（端点 nextEdge の先につながる）と、この TM\_Node とを結びつけなければならない。
2. Termination 関連は、この TM\_Node を終点にもつ TM\_Edge（端点 previousEdge の先につながる）と、この TM\_Node とを結びつけなければならない。

### 8.3.4 TM\_Edge

TM\_Edge は時間における一次元位相要素とする。

#### 関連

1. Initiation 関連は、この TM\_Edge の始点となる TM\_Node（端点 start の先につながる）と、この TM\_Edge とを結びつけなければならない。TM\_Edge は、一つのそして唯一の始点ノードをもたなければならない。
2. Termination 関連は、この TM\_Edge の終点となる TM\_Node（端点 end の先につながる）と、この TM\_Edge とを結びつけなければならない。一つの TM\_Edge は、一つのそして唯一の終点ノードをもたなければならない。

## 8.4 幾何実現

### 8.4.1 一般

応用システムが時間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 と関連をもつ（図 8-5参照）。

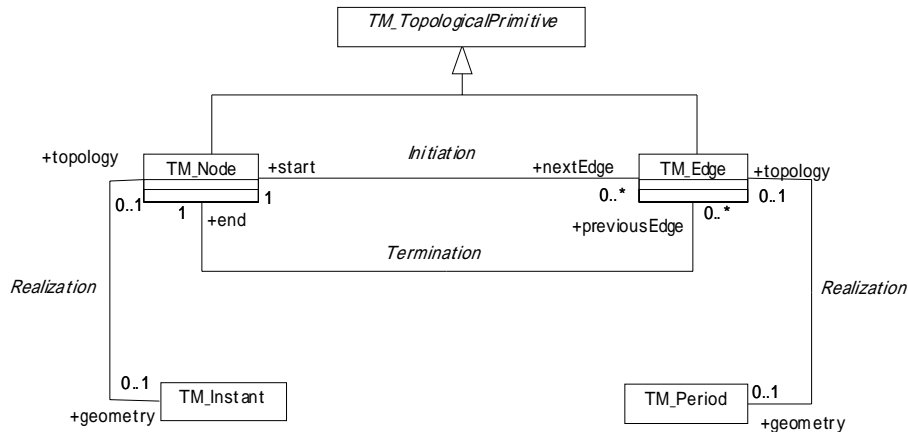


図8-5 時間位相プリミティブの幾何実現

#### 8.4.2 TM\_Node の幾何実現

TM\_Node は、Realization 関連によって対応する TM\_Instance と結びつく。この関連は、省略可能な関連である。Realization 関連の端にある geometry によって一つの TM\_Instance と関連してよいのは、一つの TM\_Node とし、この関連の端にある topology によって一つの TM\_Node と関連してよいのは、一つの TM\_Instance とする。

#### 8.4.3 TM\_Edge の幾何実現

TM\_Edge は、Realization 関連によって対応する TM\_Period と結びつく。この関連は、省略可能な関連である。Realization 関連の端にある geometry によって一つの TM\_Period と関連することができるのは唯一の TM\_Edge とし、topology によって一つの TM\_Edge と関連することができるのは唯一の TM\_Period とする。

### 8.5 時間参照系との関連

#### 8.5.1 時間参照系

時間領域における値は、時間参照系型 (TM\_ReferenceSystem) との関係の基に測定した時間上の位置とする。JIS X 0301 は、情報交換のためにグレゴリオ暦及び 24 時制地方時又は協定世界時 (UTC) の使用を規定しており、地理情報においては、これらを基本時間参照系として使用しなければならない。

#### 8.5.2 時間位置

時間位置は、時間参照系によって規定された型で特定される。地理情報を実装する際に用いる参照系は、協定世界時 (8.5.1参照) 及びグレゴリオ暦の組合せである。ISO/TS 19103 は、日付を表示するためのデータ型を JIS X 0301 に準拠する文字列として定義している。JIS X 0301 は、グレゴリオ暦及び協定世界時の使用法を定めている。JIS X 7108 では、これ以外の時間参照系で時間位置を指定するために使用しなければいけないデータ型を定義する。

#### 8.5.3 TM\_Position

TM\_Position は、その属性となる四つのデータ型のうち唯一を選択する共用体である。Date, Time, DateTime は、ISO/TS 19103 で定義している基本的なデータ型とする。これらは、日付及び時刻を文字列として符号化する JIS X 0301 に適合する。これらのデータ型は、グレゴリオ暦及び協定世界時を参照する時間位置を記述するために使用してもよい。TM\_TemporalPosition 及びその下位型は、これら以外の時間参照系を参照する時間位置を記述するために使用しなければならない。8.5.5では暦日に対応するデータ型を、8.5.6では日付を伴った時刻に対応するデータ型を規定する。これらのデータ型は、グレゴリオ暦及び協定世界時を含め、各種の暦又は時計を参照する時間位置の記述のために使用してもよい。

#### 8.5.4 TM\_TemporalPosition

TM\_TemporalPosition は、RS\_CRS との関連及び属性 indeterminatePosition をもち、二つの下位クラスが存在する。

##### 属性

1. indeterminatePosition:TM\_IndeterminateValue は、任意の属性とする。TM\_TemporalPosition の下位型を、データ型として使用しない場合、TM\_TemporalPosition は、この属性値だけを示すことになる。この属性を TM\_TemporalPosition の下位型において使用する場合、それは下位型が示す特定の時間位置情報の修飾子となる。

列挙データ型である TM\_IndeterminateValue は、不確定な位置のために次の四つの値を規定する。

“unknown” は、TM\_TemporalPosition とともに使用しなければならず、時間位置を示す値が与えられていないことを示す。

“now” は、TM\_TemporalPosition の下位型である TM\_CalDate 又は TM\_DateAndTime とともに使用しなければならず、アクセスされたときにはいつでも現時点の時間位置を TM\_CalDate で記述し値を返さなくてはならない。

“before” は、TM\_TemporalPosition の下位型である TM\_CalDate 又は TM\_DateAndTime とともに使用しなければならず、実際の時間位置は分からないが、指定した値よりも前であることを示す。

“after” は、TM\_TemporalPosition の下位型である TM\_CalDate 又は TM\_DateAndTime とともに使用しなければならず、実際の時間位置は分からないが、指定した値よりも後であることを示す。



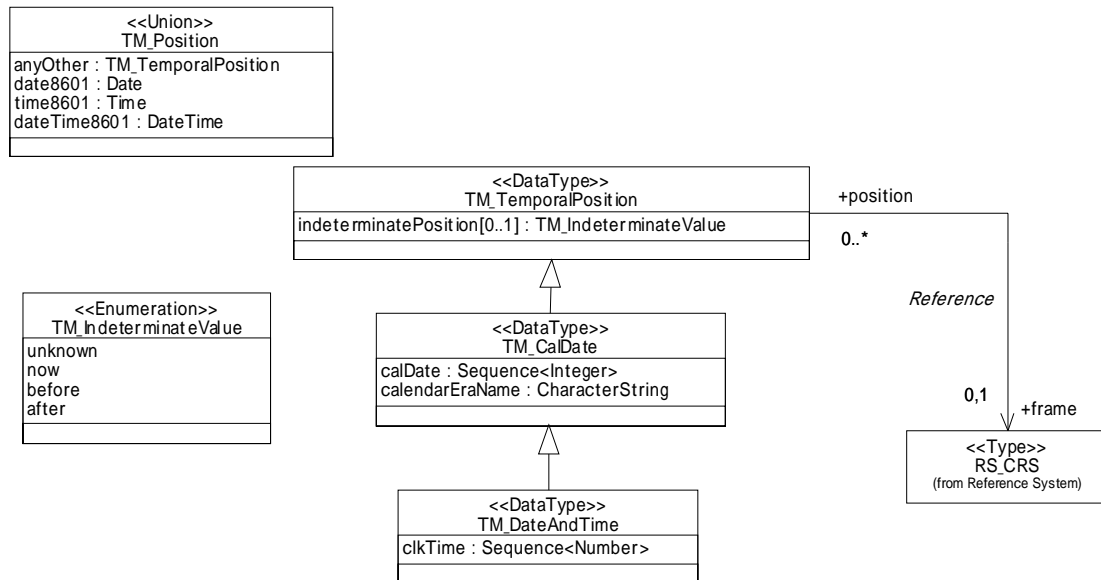


図8-6 時間位置のためのデータ型

## 関連

1. Reference 関連は、TM\_TemporalPosition を RS\_CRS（端点 frame の先につながる）に接続する。この関連は、インスタンスレベルで明示する必要はない。特に指示がない場合には、TM\_TemporalPosition は、グレゴリオ暦と協定世界時との関連を指す。また、この関連は、地物カタログの属性型定義、又はデータ集合のためのメタデータで識別することもできる。

### 8.5.5 暦日

TM\_CalDate は、ある暦の中で特定した時間位置を表すために使用しなければならないデータ型とする。

#### 属性

1. calendarEraName:CharacterString は、日付が参照する暦年代の名称を示す。
2. calDate:Sequence<Integer>は、正の整数の列で、最初の整数は、暦における階層の最上位レベルで使う単位の特定の値を示す。二番目の整数は、暦における階層の二番目のレベルで使う単位の特定の値を示し、以下同様に続く。JIS X 0301 で定義し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するあらゆる日付を示すために使用してもよい。

**例** グレゴリオ暦において、整数列“1999,09,03”は、1999年の第9月の第3日として時間位置を示す。これは、JIS X 0301 では“19990903”又は“1999-09-03”という形式で表す。

### 8.5.6 時刻を伴った日付

TM\_DateAndTime は、一日よりも短い分解能で時間位置を識別するために、TM\_CalDate を継承し、かつ時刻を示す属性をもつデータ型である。

#### 属性

1. [上位型から継承する属性]calendarEraName:CharacterString は、日付が参照する暦年代の名称を示す。
2. [上位型から継承する属性]calDate:Sequence<Integer>は、正の整数の列で、最初の整数は、

暦における階層の最上位レベルで使う単位の特定の値を示す。二番目の整数は、暦における階層の二番目のレベルで使う単位の特定の値を示し、以下同様に続く。JIS X 0301 で定義し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するあらゆる日付を示すために使用してもよい。

3. `clkTime` : Sequence <Number>は、`calDate` と類似した構造の正数の列を規定する。最初の正数は、時計の階層の中で最上位レベルに使う単位の特定の値を示し、二番目の数は二番目のレベルに使う単位の特定の値を示し、以下、同様に続く。最後の数以外は整数でなければならず、最後の数は整数又は実数のいずれでもよい。

**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 X 0301 では “221530.5” という形式で表す。

## 8.6 時間スキーマプロファイル拡張及び制限のための規則

### 8.6.1 一般

時間スキーマプロファイルを拡張及び制限する場合の規則を定める。プロファイルを拡張し、用いる場合は抽象試験項目群を設定しなくてはならない。

### 8.6.2 拡張のための規則

このプロファイルで定義したスキーマは JIS X 7108 で規定する範囲内で拡張することができる。

#### 規則

1. 時間スキーマに定義された別の型を用いたい場合は、当該クラスの上位クラスから継承させ、プロファイルに定義する。

**例** ある時間座標系の中で特定した時間位置を表すために使用しなければならないデータ型である `TM_ClockTime` を拡張したい場合は、上位クラスである `TM_TemporalPosition` から継承し、プロファイルを再定義する。

2. プロファイルに追加したいクラスの上位クラスがプロファイルに存在しない場合は上位クラスも併せてプロファイルに追加する。

**例** 時間位相複体を示す `TM_TopologicalComplex` を拡張したい場合は、その上位クラスとなる `TM_Complex` もプロファイルに存在しないため、`TM_Complex` の上位クラスである `TM_Object` から `TM_Complex` を継承させ定義し、さらに `TM_TopologicalComplex` を `TM_Complex` から継承させプロファイルに定義する。

3. 追加するクラスが必須の属性及び関連をもつ場合は、もれなく追加する。必須でないものは、目的に照らして判断する。

### 8.6.3 制限のための規則

このプロファイルで定義されたスキーマを制限し、その一部を使用することができる。

#### 規則

1. このプロファイルの一部を制限して使用する場合は、プロファイルの修正は行わず、応用スキーマに記載する。

**例** 時間位置を示す `TM_TemporalPosition` の属性 `indeterminatePosition` は、多重度が 0 又は 1 であるため、属性の省略が可能である。ある応用スキーマに存在する地物の時間属性について、時間位置を示す `TM_TemporalPosition` の属性 `indeterminatePosition` を用いないときは、プロファイルを修正し属性を削除するのではなく、応用スキーマにおいてその地物の時間属性の定義に、`indeterminatePosition` は用いないことを記載する。

## 9 被覆の幾何及び関数のためのスキーマ

**序文** このプロファイルは、ISO 19123 (Geographic information – Schema for coverage geometry and functions) から、被覆を記述するために必要となる基本的な要素を抽出及び翻訳し、プロファイルとして規定したものである。

### 9.1 スキーマの構造

ここでは、地理情報の被覆の幾何及び関数を記述する概念スキーマを示す。このプロファイルでは、被覆の定義域と値域の関係を示す関数については規定しない。

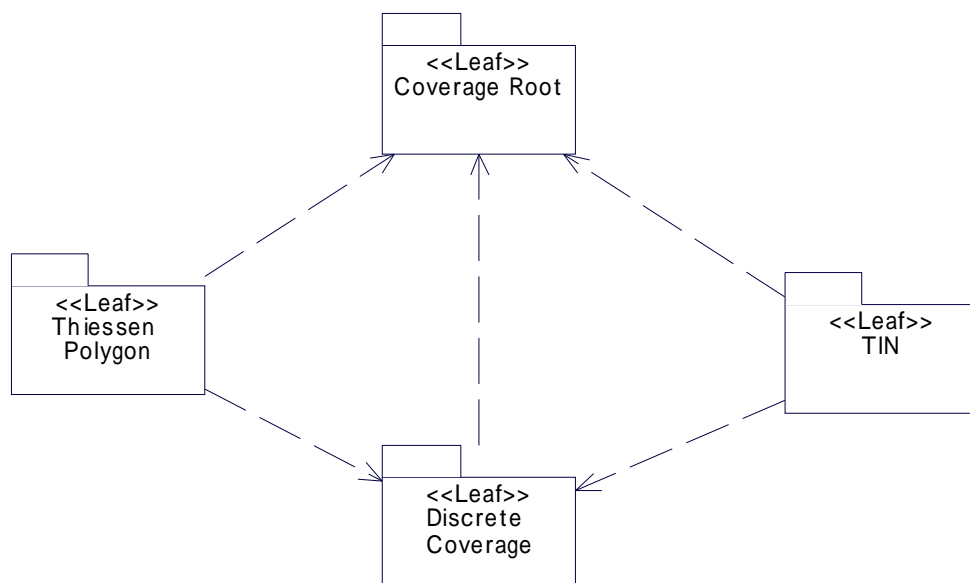


図9-1 被覆の幾何及び関数のためのスキーマプロファイルの構造

このスキーマは、四つのパッケージからなる（図 9-1参照）。被覆基底（Coverage Root）パッケージは9.2，離散被覆（Discrete Coverage）パッケージは9.3，ティーセン多角形（Thiessen Polygon）被覆パッケージは9.4，不規則三角網（TIN）被覆パッケージは9.5に記述する。

このプロファイルで定義する UML クラス名は、特定の規格を識別するため、二文字の英字及び一文字の下線からなる接頭辞から開始する。被覆の幾何及び関数のためのスキーマで規定するクラスを識別する接頭辞は、CV\_とする。

### 9.2 被覆基底パッケージ

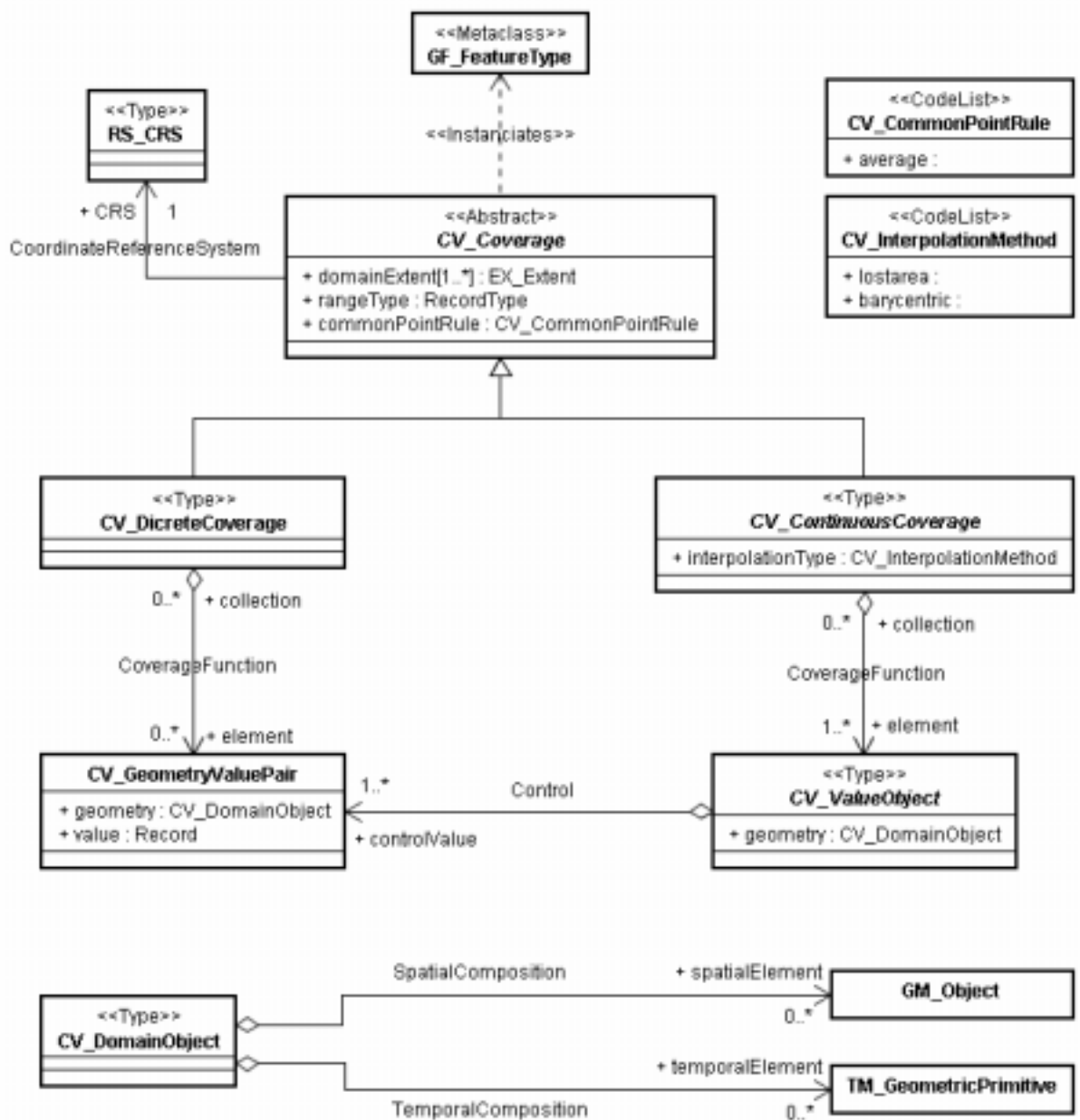


図9-2 CV\_Coverage

### 9.2.1 CV\_Coverage

CV\_Coverage は、メタクラス GF\_FeatureType (6.1.3参照) のインスタンスであり、従って地物型に相当する。CV\_Coverage は三つの属性、一つの間接をもつ (図 9-2参照)。

#### 属性

1. domainExtent[1..\*]:EX\_Extent は、被覆の時空間定義域の範囲を示す。データ型 EX\_Extent は、附属書 6 で規定している。このプロファイルが対象とする範囲は、空間範囲に限る。
2. rangeType:RecordType は、被覆の値域を記述する。データ型 RecordType は附属書 6 で規定しており、属性名とデータ型の対のリストで構成する。rangeType の最も一般的な形は属性名/データ型の対の単純なリストであるが、RecordType は再帰的に使用することができるため、より複雑な構造を表現することも可能である。特定の被覆に対する rangeType は目的に応じて応用スキーマで規定する。

3. `commonPointRule:CV_CommonPointRule` は、`CV_DomainObject` が表す被覆の幾何オブジェクトに対して、幾何オブジェクト間の境界や複数の幾何オブジェクトが重なり合う境界内に該当する位置において `CV_Coverage` の評価を行う際に用いる手続きを指定する。

**備考** この属性は、振舞いに対するパラメータであるため、JPGIS では”average”だけを定義している。

#### 関連

1. CRS 関連役割は、`CV_Coverage` を時空間定義域に含まれるオブジェクトが参照する座標参照系と関連付ける。`RS_CRS` は附属書 2 で規定している。多重度は 1 であるため、値域が同じで座標参照系が異なる時空間定義域をもつ被覆は、異なる被覆とする。

### 9.2.2 CV\_DiscreteCoverage

`CV_Coverage` の下位型である `CV_DiscreteCoverage` は、時空間定義域に含まれる単一の `GM_Object` のなかのどの直接位置に対しても同じ値を示すレコードを戻す下位型である。`CV_DiscreteCoverage` の下位型については、9.3で説明する。

#### 関連

1. `CoverageFunction` 関連は、`CV_DiscreteCoverage` を、その被覆に含まれる `CV_GeometryValuePair` の集合と関連付ける。

### 9.2.3 CV\_GeometryValuePair

`CV_GeometryValuePair` は、離散被覆の定義域と値域の関係を定義する集合の要素を記述する。このクラスの各構成要素は、被覆の時空間定義域に含まれる幾何オブジェクトと、被覆の値域に含まれる地物属性値のレコードの二つの部分で構成される。`CV_GeometryValuePair` は、幾何オブジェクトの特定の下位型と、それに対応する地物属性値レコードの組み合わせを制限するための下位型が定義される。

#### 属性

1. `geometry: CV_DomainObject` は、`CV_GeometryValuePair` の構成要素の幾何情報を `GM_Object` でもち、また、時間幾何情報を `TM_GeometricPrimitive` でもつ。
2. `value:Record` は、`GM_Object` と関連付ける地物属性値のレコードをもつ。

#### 関連

1. `CoverageFunction` 関連は、`CV_GeometryValuePair` を、自身が構成要素となる `CV_DiscreteCoverage` に関連付ける。
2. `Control` 関連は、離散被覆の場合、空になる。これについては、`CV_ValueObject` (9.2.5) で記述する。

### 9.2.4 CV\_ContinuousCoverage

`CV_ContinuousCoverage` は、時空間定義域に含まれる全ての直接位置に対して、地物属性値の個別のレコードを戻す `CV_Coverage` の下位型である。

#### 関連

1. `CoverageFunction` 関連は、この `CV_ContinuousCoverage` を、自身を構成する `CV_ValueObject` の集合と関連付ける。

### 9.2.5 CV\_ValueObject

`CV_ValueObject` は、`CV_ContinuousCoverage` 内に地物属性値を内挿するための基盤を提供する。

#### 属性

1. geometry:GM\_Object は，Control 関連によってこの CV\_ValueObject と関連付けられた CV\_GeometryValuePair の GM\_Object から構成される GM\_Object である。

### 関連

1. Control 関連は，CV\_ValueObject の作成及び CV\_ValueObject 内での CV\_GeometryValuePair の評価のための基盤を提供する CV\_GeometryValuePair の集合と，この CV\_ValueObject を関連付ける。
2. CoverageFunction 関連は，CV\_ValueObject を，自身が構成要素となる CV\_ContinuousCoverage と関連付ける。

## 9.3 離散被覆パッケージ

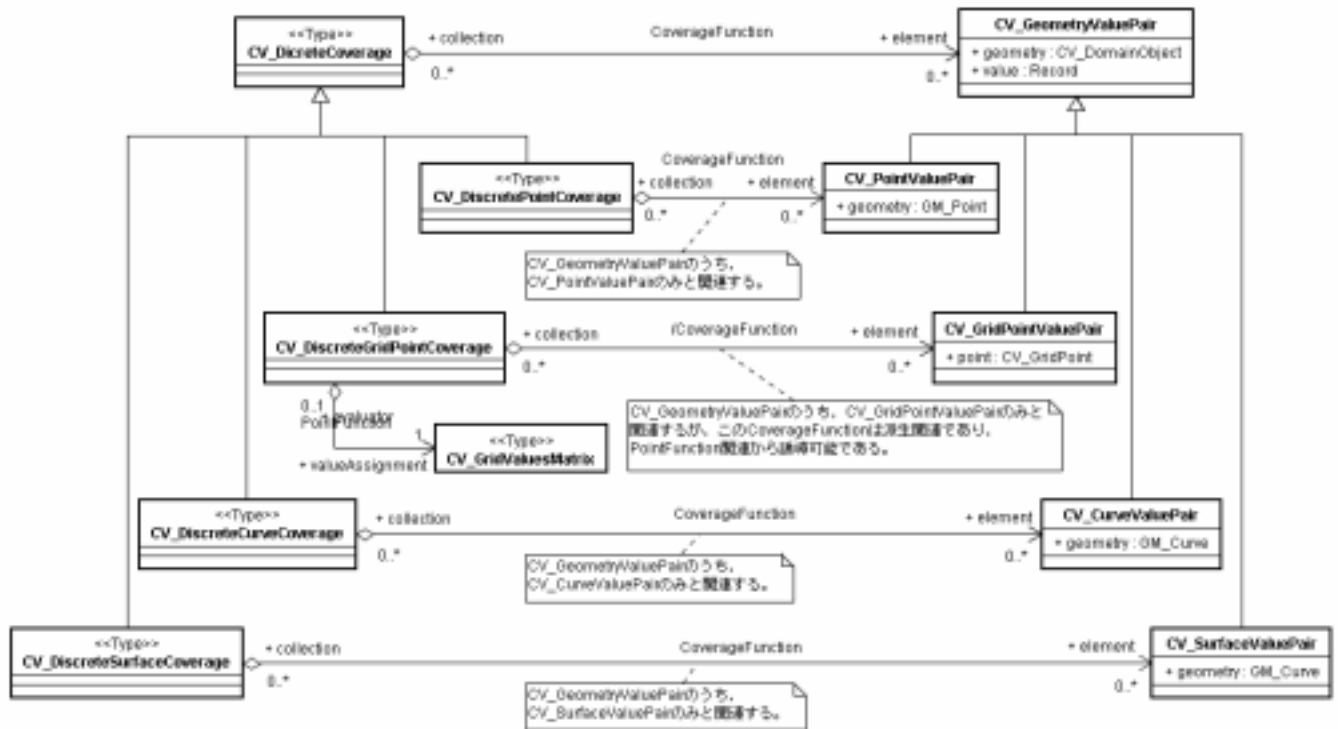


図9-3 離散被覆パッケージ

CV\_DiscreteCoverage の空間定義域は，幾何オブジェクトの集まりで構成する。CV\_DiscreteCoverage は，空間定義域における幾何オブジェクト型に基づいて下位型を規定する（図 9-3）。CV\_DiscreteCoverage の下位型は，それぞれ CV\_GeometryValuePair の下位型と関連付ける。両方のクラスの下位型は，上位クラスで指定された属性及び上位クラス間の関連を継承するが，以下（9.3.1～9.3.11）に記す制約を伴う。

**備考** JPGIS では，離散被覆は同種の幾何オブジェクトのみで構成し，互いに重なってはならない。

### 9.3.1 CV\_DiscretePointCoverage

離散点被覆は，点で構成される有限の時空間定義域によって特徴付けられる。一般に，この定義域は，不規則に分散した点の集合である。しかし，離散点被覆が主に使用されるのは，連続被覆関数に基盤を提供するためである。連続被覆関数の評価値は，離散点被覆の点の間に内挿することで得る。内挿アルゴリズムの大半は，点同士の空間的な関係の構造化されたパターンに依存

している。このため、離散点被覆の空間定義域内にある点が、規則的な方法で配列されるか、連続被覆の空間定義域が、離散点被覆の点と関係した規則的な方法で分割されなければならない。

**例** 水路測量による測深値の集合は、離散点被覆である。

#### **関連**

1. [上位の型から継承する関連]CV\_DiscretePointCoverage は、CV\_DiscreteCoverage から CoverageFunction 関連を継承するが、関連する CV\_GeometryValuePair が CV\_PointValuePair に限定されるという制約がある。

### **9.3.2 CV\_PointValuePair**

CV\_PointValuePair は、GM\_Point を幾何属性の値としてもつ CV\_GeometryValuePair の下位型である。

### **9.3.3 CV\_DiscreteCurveCoverage**

離散曲線被覆は、曲線で構成される有限の空間定義域によって特徴付けられる。この曲線は、道路や鉄道、河川などの地物を表すことが多い。これらは、ネットワークの要素となることがある。

**例** 路線番号、名称、舗装幅、舗装材料種別などを、道路を示す曲線の各部分に割り当てる被覆。

#### **関連**

1. [上位の型から継承する関連]CV\_DiscreteCurveCoverage は、CV\_DiscreteCoverage から CoverageFunction 関連を継承するが、関連する CV\_GeometryValuePair が CV\_CurveValuePair に限定されるという制約がある。

### **9.3.4 CV\_CurveValuePair**

CV\_CurveValuePair は、GM\_Curve を幾何属性の値としてもつ CV\_GeometryValuePair の下位型である。

### **9.3.5 CV\_DiscreteSurfaceCoverage**

離散曲面被覆は、定義域が曲面の集まりで構成される被覆である。多くの場合、被覆の空間定義域を構成するこの曲面は、相互に排他的である。構成要素の曲面の境界は、自然現象に対応することが多く、不規則である。

**例** 土壌種別を表す被覆は、一般に、不規則な境界をもつ曲面で構成される空間定義域をもつ。しかし、曲面被覆の空間定義域は、多角形で構成することが多い。不規則な曲面で構成される定義域と比較し多角形で構成される定義域の利点は、多角形の幾何を表現するために必要となる直接位置の数が少ないことである。どのような多角形の集合でも、離散曲面被覆に対する空間定義域として使用することができる。合同な多角形で構成される空間定義域は、広く使われている。これらの定義域は、合同な矩形や規則的な六角形で構成されることが多い。離散曲面被覆の空間定義域は TIN を構成する三角形やティーセン多角形網を構成する多角形で構成されることもある。

#### **関連**

1. [上位型から継承する関連]CV\_DiscreteSurfaceCoverage は、CV\_DiscreteCoverage から CoverageFunction 関連を継承するが、関連する CV\_GeometryValuePair の下位型が CV\_SurfaceValuePair に限定されるという制約がある。

### **9.3.6 CV\_SurfaceValuePair**

CV\_SurfaceValuePair は、GM\_Surface を幾何属性の値としてもつ CV\_GeometryValuePair の下位型である。

### 9.3.7 CV\_DiscreteGridPointCoverage

CV\_DiscreteGridPointCoverage は、被覆の範囲である四角い定義域が格子状のグリッドで分割されたグリッドセルによって構成される離散グリッド点被覆である。

この四角い定義域は、二次元グリッドであれば各辺が二つの座標軸のいずれかと平行な長方形となる。また、三次元グリッドであれば、各辺が三つの座標軸のいずれかと平行な直方体となる。四角い定義域は、各辺と平行で等間隔に配置された二本以上の線によって格子状に分割される。このとき分割する線をグリッド線、それらが交わる点をグリッド点、グリッド線間の空間をグリッドセルと呼ぶ。この格子は、グリッドの次元と同数の軸の集合として定義されたグリッド座標系のもとでグリッド軸を決め、グリッド点の相対位置をグリッド座標値として呼ぶことによって識別できる。グリッド座標値は整数値であり、グリッドセルの隅にあるグリッド点のグリッド座標値のなかで、すべてグリッド軸の座標値が最も小さな値のグリッド点、そのグリッドセルに対応したグリッド点であり、そのグリッド点の座標値を、グリッドセルを識別するための座標値として使用する。グリッド座標系を定義するためには、グリッド原点の位置、グリッド軸の方向及びグリッド線間の間隔に関する追加情報が必要であり、CV\_GridValuesMatrix がもつ属性にそれらの情報が含まれる。

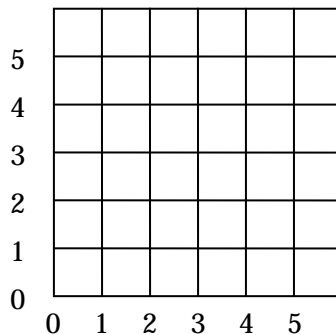


図9-4 二次元グリッドの例

**例** 地域を格子状に分割し、個々の格子面に対応した空間範囲の平均標高値を属性値とし保持した数値標高モデル (DEM) データは、離散グリッド点被覆である。同様の方法で、その格子面にその空間範囲の人口を属性値として保持した分布データも離散グリッド点被覆である。

CV\_DiscreteGridPointCoverage の場合も、他の離散被覆と同一概念を維持するため CoverageFunction 関連として CV\_GridPointValuePair の集成が定義されている。しかし、グリッド形状の特性にもとづいてデータを保持することが望ましいため、CV\_GridValuesMatrix の集成としての PointFunction 関連を定義し、CoverageFunction 関連については派生した関連として定義している。CV\_DiscreteGridPointCoverage の構成を図 9-5 に示す。

#### 関連

1. PointFunction 関連は、CV\_GridValuesMatrix が表現するグリッドで分割された個々のグリッドセルに対応した被覆の値の行列を集成する。

**備考** CV\_DiscreteGridPointCoverage が上位クラスから継承した属性値によって、グリッド全体が位置する四角い空間範囲が記述され、CV\_GridValuesMatrix が保持する属性値に





点の数は、グリッドセルの数と一致し、最もグリッド座標値が大きなグリッド点はこの対象ではない。

**例** low=(0,0), high=(99,99)と設定されていた場合、被覆の範囲は 100 x 100 のグリッドセルに分割される。このとき、グリッド線は各軸に平行に 101 本ずつあり、グリッド点は 101 x 101 個存在することになる。

### 9.3.9 CV\_GridValuesMatrix

CV\_GridValuesMatrix は、各グリッドセルに被覆の属性値を対応付ける型である。

**備考** ISO 19123 では、CV\_GridValuesMatrix は CV\_Grid の Valuation 区画部分を継承すると定義されている。JPGIS における CV\_Grid の定義は、この Valuation 区画部分に限定しているため、そのすべての属性を継承することになる。

#### 属性

1. values:Sequence<Record> は、N 個の地物属性値レコードの列である。ここで、N は、extent によって指定されたグリッド部分の中のグリッド点の数である。

**備考** 各グリッド点はグリッドセルに対応する。この values 属性は一次元の列である。

2. sequencingRule:CV\_SequenceRule は、各グリッド点を values 属性の列に対応づけるための順序付け規則を定義する。

3. startSequence:CV\_GridCoordinates は、values 属性の配列の先頭のレコードに対応するグリッド点の座標値を CV\_GridCoordinates 型で示す。

### 9.3.10 CV\_SequenceRule

CV\_SequenceRule は、二次元又は三次元の配列であるグリッド点を、一次元の配列に対応づけるための順序付け規則を指定する。

#### 属性

1. type:CV\_SequenceType は、使用する順序付け方法の型を識別する。既定値は、"linear"であり、JPGIS では"linear"だけを定義している。"linear"は、線形走査による順序付けであり、二次元グリッドの場合、図 9-6 のような順序で次元化を行う。

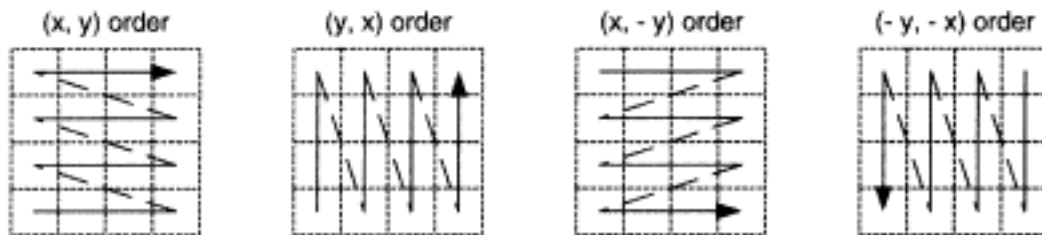


図9-6 二次元グリッドにおける線形走査の例

2. scanDirection: Sequence <CharacterString>は、axisNames 属性で定義されたグリッド座標軸の名称の前に符号を付加した文字列のリストであり、地物属性値のレコードの列内にある位置に写像されるグリッド点の順序を示す。

**例** axisNames 属性で、グリッド座標軸が" x y" と定義されているとき、scanDirection 属性で"+x -y" と設定すると、図 9-6 における (x, -y)order の順序で values 属性のレコードが並んでいることを示す。

### 9.3.11 CV\_GridPointValuePair

CV\_GridPointValuePair は、CV\_DiscreteGridPointCoverage について、他の離散被覆と同一概念で CV\_GeometryValuePair の下位型を定義したクラスである。

### 属性

1. point:CV\_GridPoint は、グリッドセルに対応した幾何属性を CV\_GridPoint データ型でもつ。この CV\_GridPoint データ型は、グリッド座標値でグリッドセルを特定する。

## 9.4 ティーセン多角形被覆パッケージ

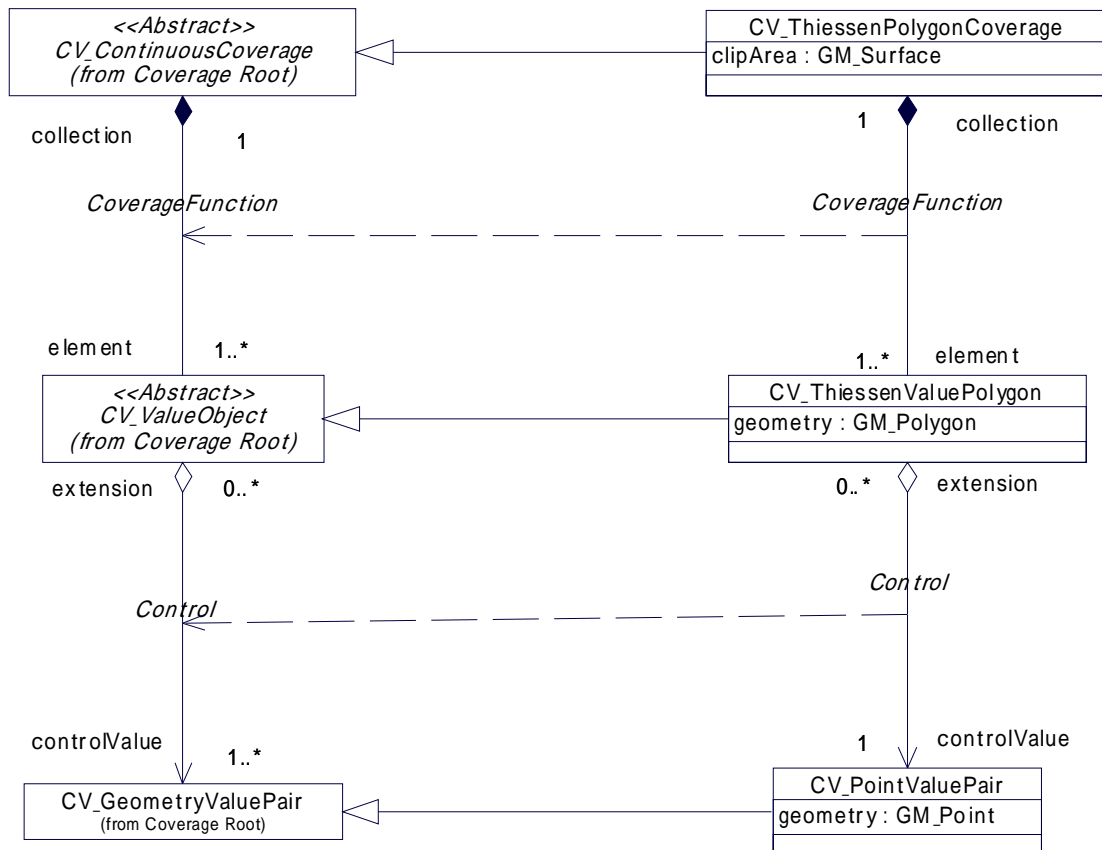


図9-7 ティーセン多角形被覆パッケージ

平面上の有限な点の集まりによって、点の集まりと同数の多角形の集まりへの平面の分割を確定することができる。ティーセン多角形は、点で構成される定義集合のうち、その定義集合に含まれる他のいかなる点よりも、その点に近い直接位置の集合を形成することによって生成する。生成された多角形には、中心と呼ばれる特殊な点が存在する。隣り合う多角形との間の境界は、それぞれの中心を結ぶ線の垂直二等分線になる。各多角形は、必ず他の一つの多角形と辺を共有する。各多角形は、定義集合から得られた点を必ず一つ含む。ティーセン多角形被覆は、ボロノイ図とも呼ばれる。

### 9.4.1 CV\_ThiessenPolygonCoverage

CV\_ThiessenPolygonCoverage は、離散点と値の集合で構成されるティーセン多角形網内の直接位置で被覆を評価する。評価は、入力位置を囲む CV\_ThiessenValuePolygon の中心間の内挿を基盤とする。

#### **属性**

1. clipArea:GM\_Surface は、CV\_ThiessenPolygonCoverage の範囲を記述する。その境界は、網目の最も外側にある多角形の境界を確定するものである。

#### **関連**

1. [上位型から継承する関連]CoverageFunction 関連は、上位型である CV\_ContinuousCoverage から継承するが、以下のようにオーバーライドする。すなわち、CV\_ThiessenPolygonCoverage を、自身を構成する CV\_ThiessenValuePolygon と関連付ける。

### **9.4.2 CV\_ThiessenValuePolygon**

CV\_ThiessenValuePolygon は、CV\_ValueObject の下位型である。

#### **属性**

1. [上位型から継承する属性]geometry:GM\_Polygon は、上位型である CV\_ValueObject から継承するが、以下のようにオーバーライドする。すなわち、Control 関連によって識別された CV\_PointValuePair を中心とするティーセン多角形の幾何 (GM\_Polygon) を保持する。

#### **関連**

1. [上位型から継承する関連]Control 関連は、上位型である CV\_ValueObject から継承するが、以下のようにオーバーライドする。すなわち、CV\_ThiessenValuePolygon を、その中心で一つの CV\_PointValuePair と関連付ける。
2. [上位型から継承する関連]CoverageFunction 関連は、上位型である CV\_ValueObject から継承するが、以下のようにオーバーライドする。すなわち、CV\_ThiessenValuePolygon を、自身が構成要素となる CV\_ThiessenPolygonCoverage と関連付ける。

### **9.5 不規則三角網被覆パッケージ**

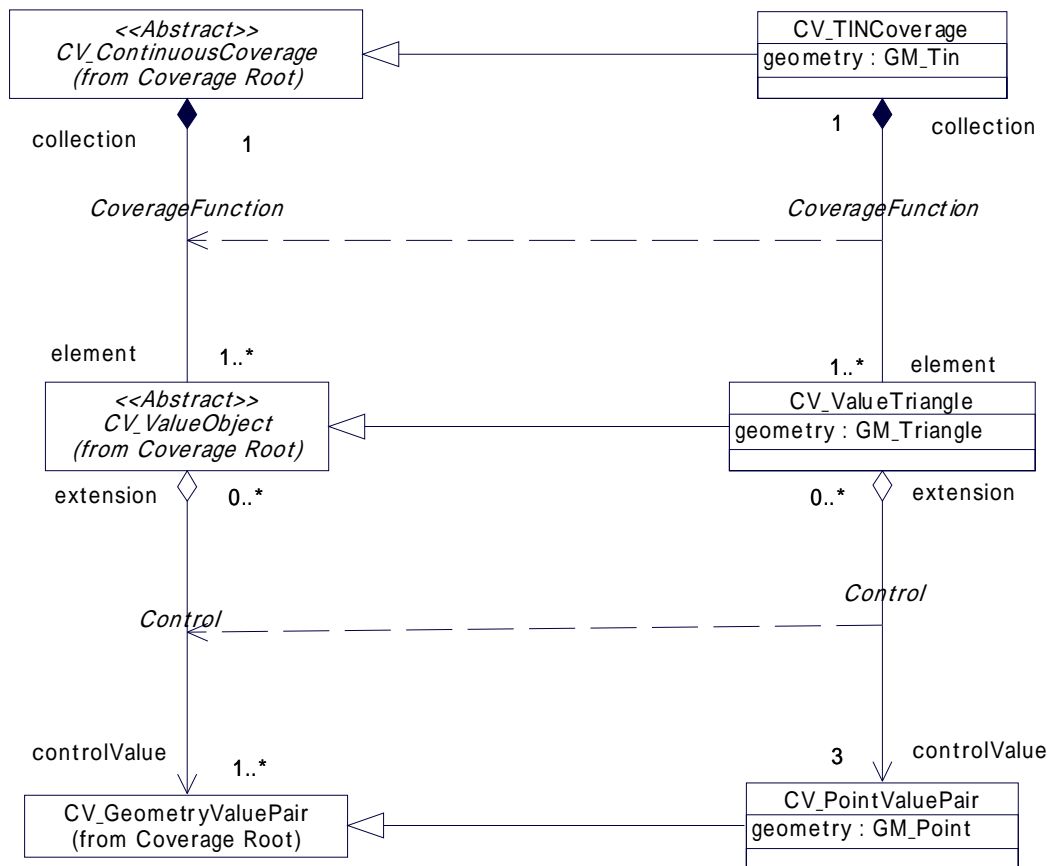


図9-8 不規則三角網被覆パッケージ

不規則三角網（TIN）の基本的な考え方は、離散点被覆の空間定義域に含まれる点の集合の凸包を、演算上一意となる互いに辺で隣接する三角形の集合に分割するというものである。個々の三角形は、離散点被覆の空間定義域に含まれる三つの点で形成する。ドロネ三角分割法は、最適等角の形状を有し、個々の三角形の外接円が、その三角形の頂点以外の離散点被覆の点をもたないような三角形で構成する TIN 分割を作成するために一般的に用いられる。GM\_Tin (7.2.3.13) は、ドロネ三角分割を示す。

### 9.5.1 CV\_TINCoverage

CV\_TINCoverage は、GM\_Tin によって特徴付けられた CV\_ContinuousCoverage の下位型である。その属性値は、それぞれの隅に与えられる属性値のレコードを用い、分割内の個々の三角形において内挿を行うことにより計算する。ただし、このプロファイルでは内挿法を規定しない。

#### 属性

1. geometry:GM\_Tin は、この被覆を評価するための構造を提供する不規則三角網を保持する。

#### 関連

1. [上位型から継承する関連]CoverageFunction 関連は、上位型である CV\_ContinuousCoverage から継承するが、以下のようにオーバーライドする。すなわち、この CV\_TINCoverage を、自身を構成する CV\_ValueTriangle と関連付ける。

### 9.5.2 CV\_ValueTriangle

CV\_ValueTriangle は、GM\_Point が共線ではない三つの CV\_PointValuePair で構成される CV\_ValueObject の下位型である。

#### 属性

1. [上位型から継承する属性]geometry:GM\_Triangle は、三つの CV\_PointValuePair の位置を定義する GM\_Triangle を保持する。

#### 関連

1. [上位型から継承する関連]Control 関連は、上位型である CV\_ValueObject から継承するが、以下のようにオーバーライドする。すなわち、この CV\_ValueTriangle を、その頂点で三つの CV\_PointValuePair と関連付く。
2. [上位型から継承する関連]CoverageFunction 関連は、上位型である CV\_ValueObject から継承するが、以下のようにオーバーライドする。すなわち、CV\_ValueTriangle を、自身が構成要素となる CV\_TINCoverage と関連付ける。

### 9.6 空間スキーマとの関連

被覆の空間定義域は、直接位置により記述される幾何オブジェクトの集合である。この幾何オブジェクトは、7.2.1.1で規定する GM\_Object である。

被覆要素と空間スキーマは、被覆要素の定義域に GM\_Object 又はその下位型を示すことで関係付けられる。

### 9.7 被覆の幾何及び関数のためのスキーマプロファイル拡張及び制限のための規則

#### 9.7.1 一般

被覆の幾何及び関数のためのスキーマプロファイルを拡張及び制限する場合の規則を定める。プロファイルを拡張し、用いる場合は抽象試験項目群を設定し、適合性試験に合格しなくてはならない。

#### 9.7.2 拡張のための規則

このプロファイルで定義したスキーマを拡張し、使用することができる。

#### 規則

1. このプロファイルで定義しておらず、原規格で定義している別の型を用いたい場合は、当該クラスの上位クラスから継承させ、プロファイルに定義する。

**例** 六角形グリッド被覆を示す CV\_HexagonalGridCoverage を使用したい場合は、上位クラスである GM\_ContinuousCoverage から継承し、プロファイルを再定義する。

2. このプロファイルに追加したいクラスの時空間定義域が GM\_Object で、空間スキーマプロファイルにおいて定義していない場合、各プロファイルの拡張のための規則に従い、各プロファイルに要素を定義する。

**例** 離散立体被覆を示す CV\_DiscreteSolidCoverage を使用したい場合は、空間定義域が GM\_Solid であるため、このプロファイルに CV\_DiscreteSolidCoverage を追加定義すると共に、空間スキーマプロファイルに GM\_Solid を追加定義する。

3. 追加するクラスが必須の属性及び関連をもつ場合は、もれなく追加する。必須でないものは、目的に照らして判断する。

#### 9.7.3 制限のための規則

このプロファイルで定義されたスキーマを制限し、その一部を使用することができる。

**規則**

1. 被覆の幾何及び関数のためのスキーマプロファイルの一部を制限して使用する場合は、プロファイルの修正は行わず、応用スキーマに記載する。

**例** CV\_Coverage のうち、下位型である CV\_DiscreteSurfaceCoverage を用いないときは、プロファイルを修正し CV\_DiscreteSurfaceCoverage を削除するのではなく、応用スキーマにおいて CV\_DiscreteSurfaceCoverage は用いないことを記述する。

## 10 地理識別子による空間参照

**序文** このプロファイルは、ISO 19112 (Geographic information - Spatial referencing by geographic identifiers) を翻訳し、技術的内容及び規格票の様式を変更することなく作成した JIS X 7112 (地理情報—地理識別子による空間参照) から、地理識別子による空間参照を記述するために必要となる基本的な要素を抽出し、プロファイルとして定義したものである。

### 10.1 地理識別子による空間参照の概念

#### 10.1.1 地理識別子による空間参照

地物の位置は、空間参照によって特定する。地理識別子による空間参照は、一つの場所を一意に特定する。場所とは、ここでは他の地物を空間参照するために使用する地物とする。地理識別子による空間参照を用いるには、地理識別子によって特定される場所の型を定義しなければならない。

**備考** 地理データ集合の地物の空間参照は、通常は地物の属性として保持され、位置との関係を定義する。位置との関係は通常、包含である。しかし、“隣接する”とか、識別されている位置から測定した距離や方向とともに用いた“～沿いの距離”というような関係を用いて、より複雑な空間参照を構成することもある。道路及び鉄道に対する参照系は、リンク（道路又は路線）に沿う一つのノード（端点又は交差点）からの測定距離に基づくことが多い。データ集合が使用する空間参照系は、JMP2.0 で定義するように、データ集合のメタデータの一部を構成する。

#### 10.1.2 地理識別子による空間参照系

地理識別子による空間参照系は、対応する地理識別子をもつ一つ以上の、相互に関連する場所型の集合からなる。これらの場所型は、集成又は分割によって、場合によっては階層構造を形成しながら、互いに関連することができる。

#### 10.1.3 地名辞典

地名辞典は、場所インスタンスを記述する地理識別子の要覧とする。地名辞典は、個々の場所インスタンスの位置に関する追加情報 (SI\_Gazetteer の属性 scope に記述することができる) を含むことがある。ここには、座標参照を含むこともあるが、純粹に記述的なものでもよい。座標参照を含む場合、地理識別子による空間参照系から座標参照系へ変換することも可能である。記述的な参照 (SI\_Gazetteer の属性 geographicExtent に記述することができる) を含む場合、これは、例えば地所の郵便番号など、地理識別子を伴う別の空間参照系を使った空間参照となる。どの場所型についても、一つ以上の地名辞典に含まれる可能性がある。

### 10.2 地理識別子による空間参照スキーマ

このプロファイルで定義する UML クラス名は、特定の規格を識別するため、二文字の英字及び一文字の下線からなる接頭辞から開始する。地理識別子による空間参照で規定するクラスを識別する接頭辞は、SI\_とする。



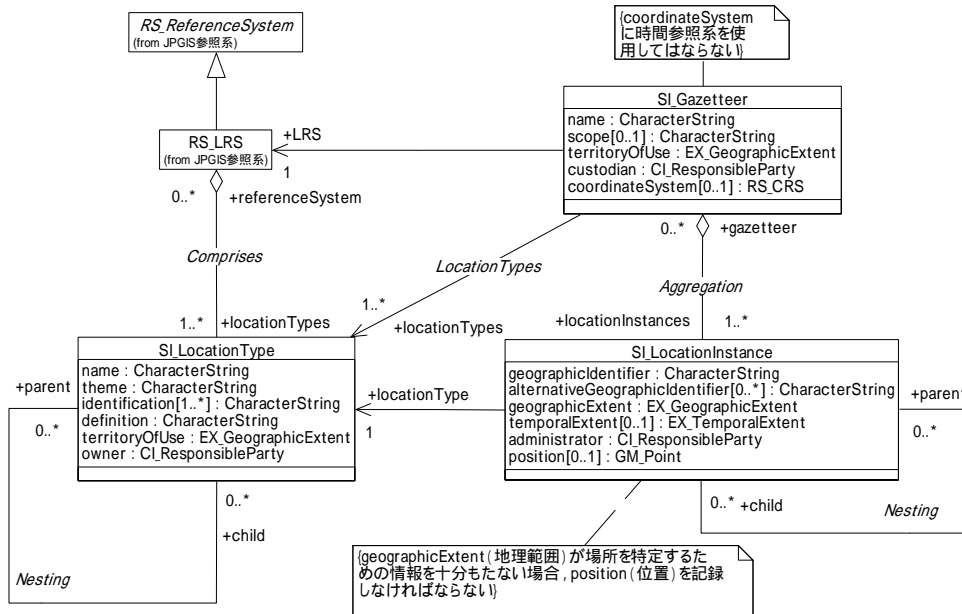


図10-1 地理識別子による空間参照スキーマ

### 10.2.1 SI\_LocationType

空間参照系のそれぞれの場所型 (SI\_LocationType) は、次の属性を必要とする。また、場所型のいずれかの属性が変化した場合、場所型の新しいバージョンを生成しなければならない。

**属性** SI\_LocationType は、次の属性を必要とする。

1. 名前 (name:CharacterString) は、場所型の名前を示さなければならない。
2. 主題 (theme:CharacterString) は、場所型の特徴を記述しなければならない。
3. 識別 (identification[1..\*]:CharacterString) は、場所インスタンスを特定するための方法を示さなければならない。記述例は、“名前”，“コード”である。
4. 定義 (definition:CharacterString) は、場所型の定義を記述しなければならない。

**備考** 場所型の定義は、次のひとつの形態に含まれるものとする。

- 境界の集まりで定義される領域。例えば、境界線で定義される国。
- 単一地物。例えば、中心線で定義される街路、又はそのような二本の街路の交差点。
- より小さな地物の集まり。例えば、構成国で定義される貿易圏。

場所型がより小さな単位の集まりで定義されているときは、その境界は厳密でなくてもよい。例えば、郵便配達点の集まりで定義された郵便番号。

5. 使用地域 (territoryOfUse:EX\_GeographicExtent) は、附属書 6 で定義している EX\_GeographicExtent を型として、場所型を使用する地域を示さなければならない。
6. 責任者 (owner:CI\_ResponsibleParty) は、附属書 6 で定義している CI\_ResponsibleParty を型として、場所インスタンスの作成・破棄ができる組織の名前を示さなければならない。

### 関連

1. Nesting 関連は、この SI\_LocationType と、その一部分である他の SI\_LocationType, 又はこの SI\_LocationType を細分化する他の SI\_LocationType とを関連付けることで、場所型同士で集成又は分割によって、場合によっては階層構造を形成することを可能としている。
2. Comprises 関連は、RS\_LRS と、この SI\_LocationType を関連付ける集成とする。

### 10.2.2 SI\_Gazetteer

地名辞典 (SI\_Gazetteer) は、空間参照系のなかにある場所型を参照する場所インスタンスの要覧とする。場所インスタンスが作成又は廃止されたときや、場所インスタンスの新しいバージョンが作成されたときは、地名辞典の新しいバージョンを作成しなければならない。

**備考** 一つの地名辞典に対応する複数の異なる場所型が存在でき、その場所インスタンスは、さまざまな方法で識別される。逆に、一つの場所型は、複数の地名辞典から参照されることもある。

#### 属性

1. 名前 (name:CharacterString) は、地名辞典の名前を示さなければならない。地名辞典の基準日は、名前に含まれる。
2. 適用範囲 (scope[0..1]:CharacterString) は、地名辞典に含まれた場所型について記述することができる。この属性は任意の属性とする。
3. 使用領域 (territoryOfUse:EX\_GeographicExtent) は、附属書 6 で定義している EX\_GeographicExtent を型として、地名辞典の収録地域を示さなければならない。
4. 責任者 (custodian:CI\_ResponsibleParty) は、附属書 6 で定義している CI\_ResponsibleParty を型として、地名辞典のメンテナンスに責任がある組織の名称を示さなければならない。
5. 座標参照系 (coordinateSystem[0..1]:RS\_CRS) は、附属書 2 で定義している RS\_CRS を型として、地名辞典の中で位置を記述するために使われる座標系の名前を示すことができる。この属性は任意の属性とする。

#### 関連

1. Aggregation 関連は、この SI\_Gazetteer と SI\_LocationInstance とを関連付ける集成であり、SI\_Gazetteer は一つ以上の SI\_LocationInstance で構成しなければならない。
2. locationTypes 関連役割は、この SI\_Gazetteer を SI\_LocationType に関連付ける。SI\_Gazetteer は一つ以上の SI\_LocationType を参照しなければならない。
3. LRS 関連役割は、RS\_LRS と、この SI\_Gazetteer とを関連付ける。SI\_Gazetteer は一つの RS\_LRS を参照しなければならない。

### 10.2.3 SI\_LocationInstance

場所インスタンス (SI\_LocationInstance) は、空間参照系のなかにある場所型を参照するインスタンスである。いずれかの属性が変化した場合、SI\_LocationInstance の新しいインスタンスを生成しなければならない。ただし、この場合は、属性 temporalExtent を記述しなければならない。SI\_LocationInstance の変化の例としては、境界変更によって地方自治体の行政区域が変わったときがある。

#### 属性

1. 地理識別子 (geographicIdentifier:CharacterString) は、地名辞典の中で一意となるインスタンスの識別子を示さなければならない。
2. 代替地理識別子 (alternativeGeographicIdentifier[0..\*]:CharacterString) は、geographicIdentifier で記述した以外の、この場所インスタンスに対する別の地理識別子を記述することができる。この属性は任意の属性とする。
3. 地理範囲 (geographicExtent:EX\_GeographicExtent) は、附属書 6 で定義している EX\_GeographicExtent を型として、場所インスタンスの範囲を示さなければならない。
4. 時間範囲 (temporalExtent[0..1]:EX\_TemporalExtent) は、附属書 6 で定義している EX\_TemporalExtent を型とする。この属性は任意の属性のため、使用しなくてもよい。もし使用する場合は、この場所インスタンスの存続期間を表す。

5. 管理者 (administrator:CI\_ResponsibleParty) は、附属書 6 で定義している CI\_ResponsibleParty を型として、場所インスタンスの特性を定義する責任がある組織の名前を示さなければならない。
6. 位置 (position[0..1]:GM\_Point) は、7.2.2.6 で定義している GM\_Point を型として、場所インスタンスの代表点の座標を示す。これによって座標による空間参照の仕組みに関連付けることができる。

#### 関連

1. Nesting 関連は、この SI\_LocationInstance をその一部分である他の SI\_LocationInstance (端点 parent の先につながる)、又はこの SI\_LocationInstance を細分化する他の SI\_LocationInstance (端点 child の先につながる) に関連付ける。
2. locationType 関連役割は、この SI\_LocationInstance を SI\_LocationType に関連付ける。SI\_LocationInstance は SI\_LocationType を参照しなければならない。
3. Aggregation 関連は、SI\_Gazetteer と、この SI\_LocationInstance を関連付ける集成とする。

#### 制約

1. geographicExtent が場所を特定するための情報を十分持たない場合 (つまり、型として EX\_GeographicDescription を用いた場合) は、属性 position を記録しなければならない。

### 10.3 地理識別子による空間参照プロファイル拡張及び制限のための規則

#### 10.3.1 一般

地理識別子による空間参照プロファイルを拡張及び制限する場合の規則を定める。プロファイルを拡張し用いる場合は、原規格を参照し抽象試験項目群を設定しなければならない。

#### 10.3.2 拡張のための規則

##### 規則

1. このプロファイルで定義していない要素を追加する場合又は定義域を拡張する場合は、プロファイルを再定義する。

**例** SI\_LocationInstance の temporalExtent 属性を必須の属性としたい場合は、プロファイルを拡張する。

#### 10.3.3 制限のための規則

このプロファイルで定義されたスキーマの一部を制限し、使用することができる。

##### 規則

1. 地理識別子による空間参照プロファイルの一部を制限して使用する場合は、プロファイルの修正は行わず、応用スキーマに記載する。

**例** SI\_LocationType 内での階層構造は必要ないと考え、SI\_LocationType の Nesting 関連を用いないとした場合は、プロファイルを修正し関連を削除するのではなく、応用スキーマにおいてその地物の時間属性の定義に、SI\_LocationType の Nesting 関連は用いないことを記載する。

## 11 地物カタログ化法

**序文** このプロファイルは、ISO 19110 (Geographic information – Methodology for feature cataloguing) から、地物カタログを作成するために必要となる基本的な規則を抽出及び翻訳し、プロファイルとして定義したものである。

### 11.1 一般

このプロファイルは、地物型のカタログ化のための方法を規定する。このプロファイルは、地物型の分類を地物カタログの中で体系化し、地理データ集合の使用者に提供する方法を規定する。このプロファイルは、まだカタログ化されていない分野の地物型のカタログを作成したり、既存の地物カタログをこのプロファイルに従ったものに修正したりするために適用することができる。このプロファイルは、デジタル形式で表現される地物型のカタログ化に適用する。

このプロファイルは、型レベルでの地物の定義に適用することができ、それぞれの型の個々のインスタンスの表現には適用することができない。また、このプロファイルで規定する空間スキーマ、時間スキーマ、附属書 10 の描画法及び地物インスタンス収集のための基準は除外する。

### 11.2 主要な要件

#### 11.2.1 地物カタログ

地物カタログは、実世界の現象の一つの明確な分類として、一つ以上の地理データ集合において表現される実世界の抽象概念を表す。地物カタログにおける基本レベルの分類は地物型である。地物カタログは、地物を含むどのような地理データに対しても、デジタル形式で使用可能でなければならない。地物カタログは、どのような既存の地理データ集合とも独立して、このプロファイルに準拠することができる。

#### 11.2.2 カタログ要素

##### 11.2.2.1 一般

以下の項では、地物カタログ要素に対する一般の又は特殊な要件を示す。ここで地物カタログ要素とは、地物型、地物属性、地物関連及び関連役割の四つを指す。

##### 11.2.2.2 完全性

11.3.2において、地物分類情報の表現のためのテンプレートを規定する。このテンプレートに従って作成される地物カタログは、与えられた地理データ集合に含まれるすべての地物型を文書化しなければならない。地物カタログは、11.3.2の 1 (FC\_FeatureCatalogue) において規定する識別情報を含まなければならない。地物カタログは、地理データ集合に含まれるすべての地物型と、個々の地物型がもつ地物属性、地物関連及び関連役割の定義及び記述を含まなければならない。異なる応用システム間の地物カタログの内容の予測可能性及び比較可能性を保証するために、地物カタログが、11.3.2に規定する要素のみを含むことを推奨する。また、異なる応用システム間の地物カタログの実用性を最大化するために、地物カタログ情報をモデル化するにあたり、概念スキーマ言語を使用することを推奨する。

##### 11.2.2.3 一般要件

#### 11.2.2.3.1 名前の形式

地物カタログに含まれるすべての地物カタログ要素は、自身が含まれる地物カタログにおいて一意となる名前によって識別されなければならない。地物カタログにおいて、同一名称の地物カタログ要素が複数出現する場合は、全ての出現においてその定義が同じでなければならない。

#### 11.2.2.3.2 定義属性の形式

地物カタログに含まれる地物カタログ要素の定義属性は、自然言語で記述されなければならない。これらの定義属性は、カタログが別の定義元資料を指定していない限り、一つのカタログに含まなければならない。定義元資料と地物カタログにおいて同じ用語が出現する場合は、地物カタログにおける定義属性を優先しなければならない。

#### 11.2.2.4 地物型の要件

各地物型は一つの名称で識別されなければならない。その定義属性は自然言語で記述しなければならない。各地物型は、複数の別名をもつことができる。また地物カタログには、各地物型に対し、必要に応じて地物属性、地物関連、関連役割を含まなければならない。

#### 11.2.2.5 地物属性の要件

地物属性は必要に応じ、各地物型に対して特定し、定義しなければならない。その定義は、自然言語による定義属性及び属性値のデータ型からなる。

#### 11.2.2.6 地物属性定義域の要件

地物属性定義域は必要に応じ、各地物属性に対してラベルを付さなければならない。ラベルは、リスト中の値であり、地物属性において一意でなければならない。

#### 11.2.2.7 地物関連の要件

地物関連は必要に応じ、名前をつけ、定義されなければならない。関連の名前と関連に係わる地物型の役割は特定されなければならない。

#### 11.2.2.8 関連役割の要件

関連役割は必要に応じ、名前をつけ、定義されなければならない。地物関連に係わり、地物関連及び役割をもつ地物型の名称は特定されなければならない。

### 11.3 地物カタログのテンプレート

#### 11.3.1 地物カタログの概念モデル

ここでは、地物カタログの概念モデルを示す。

このプロファイルで定義する UML クラス名は、特定の規格を識別するため、二文字の英字及び一文字の下線からなる接頭辞から開始する。地物カタログで規定するクラスを識別する接頭辞は、FC\_とする。

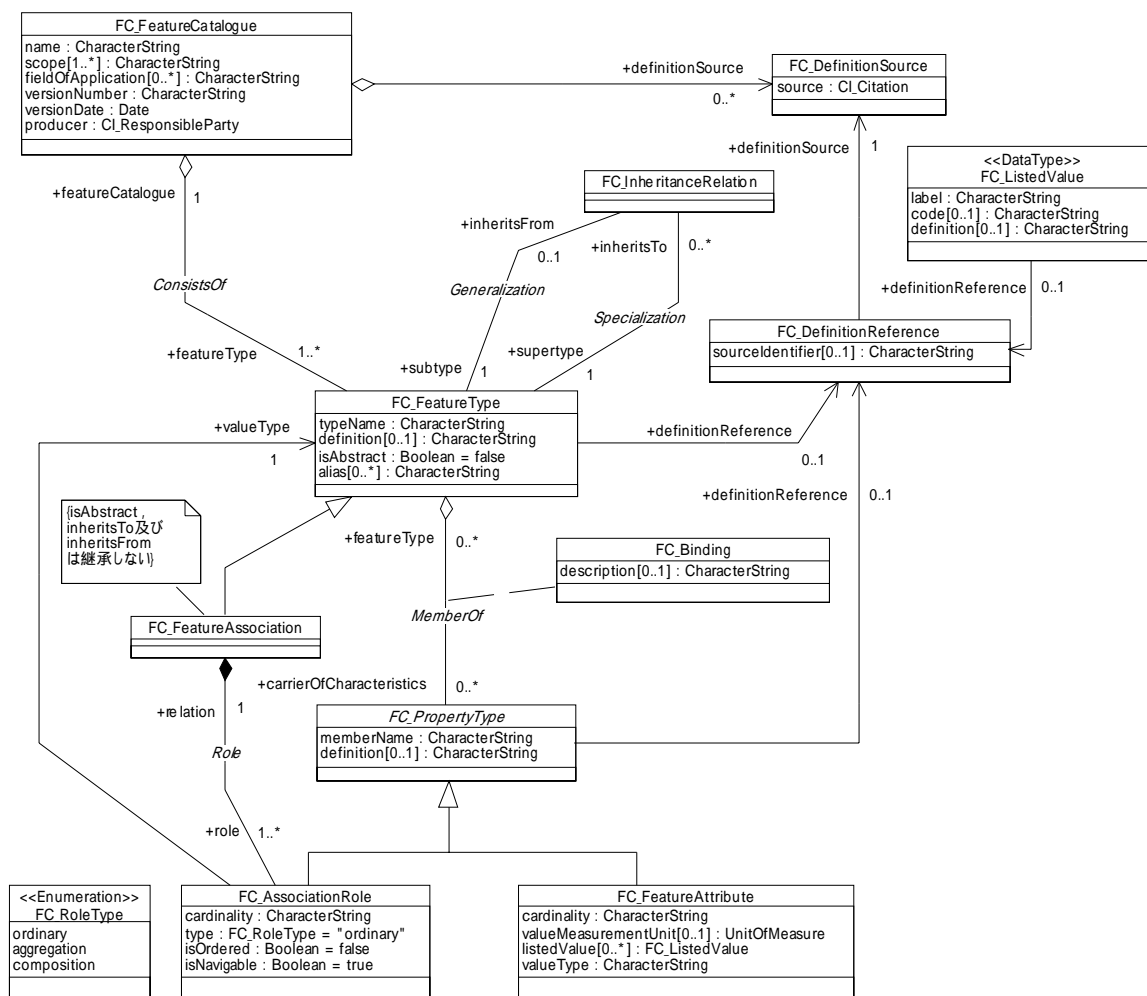


図11-1 地物カタログの概念モデル

図 11-1は地物カタログの概念モデルである。各要素の説明は、11.3.2地物カタログのテンプレートに示す。

### 11.3.2 地物カタログのテンプレート

ここでは、このプロファイルで規定する地物カタログの情報を示す。

地物カタログの要素情報の必須又は任意の区別は、以下に示す表記法を使用する。

M: その要素が必須であり、地物カタログに含まなければならないことを示す。

O: その要素が任意であることを示す。その要素を地物カタログに含む場合は、その要素の必須のサブ要素を含まなければならない。

表11-1 地物カタログテンプレート

	名称	記述	必須/任意	多重度 最大値	型	制約
1	【クラス】 FC_FeatureCatalogue	識別、問合せ情報、必要な情報をもついくつかの地物型の定義を含んだ地物カタログ。				
1.1	【属性】 name	地物カタログの名称。	M	1	CharacterString	
1.2	【属性】 scope	地物カタログに定義された地物型の主題領域。	M	N	CharacterString	
1.3	【属性】 fieldOfApplication	地物カタログの使用法の種類の記述。	0	N	CharacterString	
1.4	【属性】 versionNumber	地物カタログの版番号。	M	1	CharacterString	
1.5	【属性】 versionDate	地物カタログの発効日。	M	1	Date	
1.6	【属性】 producer	地物カタログの知的財産権を所有する主な責任主体となる個人又は組織の名称、住所、国、連絡先。	M	1	CI_ResponsibleParty	
1.7	【関連役割】 featureType	地物カタログに含まれる地物型への関連。	M	N	FC_FeatureType	集成
1.8	【関連役割】 definitionSource	地物カタログに含まれる地物型及び地物プロパティの定義の情報源への関連。	0	N	FC_DefinitionSource	集成
2	【クラス】 FC_FeatureType	共通な特性をもつ実世界の現象を表す型。				
2.1	【属性】 typeName	地物カタログ内で地物型を一意に識別するための文字列。	M	1	CharacterString	
2.2	【属性】 definition	自然言語で記述する地物型の定義。	0	1	CharacterString	
2.3	【属性】 isAbstract	地物型が抽象型かどうかの区別。	M	1	Boolean	
2.4	【属性】 alias	地物型の名前と同等の名前。	0	N	CharacterString	
2.5	【関連役割】 inheritsFrom	関連等の特性を継承する上位クラスへの関連。	0	1	FC_InheritanceRelation	
2.6	【関連役割】 inheritsTo	関連等の特性を継承させる下位クラスへの関連。	0	N	FC_InheritanceRelation	
2.7	【関連役割】 featureCatalogue	地物型が含まれる地物カタログへの関連。	M	1	FC_FeatureCatalogue	
2.8	【関連役割】 carrierOfCharacteristics	地物型に含まれる地物プロパティへの関連。	0	N	FC_PropertyType	集成
2.9	【関連役割】 definitionReference	地物型の定義情報源への関連。	0	1	FC_DefinitionReference	
3	【クラス】 FC_InheritanceRelation	GF_InheritanceRelationを実現するFC_InheritanceRelation。				
3.1	【関連役割】 subtype	地物型の特化の関係。	M	1	FC_FeatureType	
3.2	【関連役割】 supertype	地物型の汎化の関係。	M	1	FC_FeatureType	
4	【クラス】 FC_PropertyType	地物プロパティの抽象型。				
4.1	【属性】 memberName	地物型に含まれる地物プロパティの名前。	M	1	CharacterString	
4.2	【属性】 definition	自然言語で記述する地物プロパティの定義。	0	1	CharacterString	
4.3	【関連役割】 featureType	地物プロパティを含む地物型への関連。	0	N	FC_FeatureType	

4.4	【関連役割】 definitionReference	定義の情報源への関連。	0	1	FC_DefinitionReference	
5	【クラス】 FC_Binding	プロパティが特定の地物型に結びつく方法を記述する型。地物型とプロパティ型に定義されている MemberOf 関連に対する関連型として使用される。				
5.1	【属性】 description	プロパティが特定の地物型にどのように結びつくかの記述。	0	1	CharacterString	
6	【クラス】 FC_FeatureAttribute	地物型の特性。				
6.1	【属性】 memberName	地物型に含まれる地物属性の名前。	M	1	CharacterString	
6.2	【属性】 definition	自然言語で記述する地物属性の定義。	0	1	CharacterString	
6.3	【属性】 cardinality	地物型における地物属性の個数。	M	1	CharacterString	
6.4	【属性】 valueMeasurementUnit	地物属性で使用される単位。	0	1	UnitOfMeasure	
6.5	【属性】 listedValue	地物属性に許容される値。存在する場合は、地物属性の型は列挙型又は符号リスト型である。	0	N	FC_ListedValue	
6.6	【属性】 valueType	地物属性値の型。	M	1	CharacterString	
6.7	【関連役割】 featureType	地物プロパティを含む地物型への関連。	0	N	FC_FeatureType	
6.8	【関連役割】 definitionReference	定義の情報源への関連。	0	1	FC_DefinitionReference	
7	【クラス】 FC_AssociationRole	地物関連の役割。				
7.1	【属性】 memberName	地物型に含まれる関連役割の名前。	M	1	CharacterString	
7.2	【属性】 definition	自然言語で記述する関連役割の定義。	0	1	CharacterString	
7.3	【属性】 cardinality	一つの関連相手の地物インスタンスに対して役割となる地物インスタンスの数。	M	1	CharacterString	
7.4	【属性】 type	関連役割の種類。関連、集成、合成の区別。	M	1	FC_RoleType	初期値 = 1 ("ordinary")
7.5	【属性】 isOrdered	関連役割のインスタンスに順序があるかどうかの区別。false は順序がなく、true は順序がある。	M	1	Boolean	初期値 = false
7.6	【属性】 isNavigable	関連の関連元地物から関連先地物に対して役割が参照できるかどうかの区別。	M	1	Boolean	初期値 = true
7.7	【関連役割】 featureType	地物型への関連。	0	N	FC_FeatureType	
7.8	【関連役割】 definitionReference	定義の情報源への関連。	0	1	FC_DefinitionReference	
7.9	【関連役割】 relation	関連役割が部分となる関連。	M	1	FC_FeatureAssociation	
7.10	【関連役割】 valueType	関連役割の相手の型。	M	1	FC_FeatureType	
8	【クラス】 FC_ListedValue	列挙型地物属性の値。コード及び解釈を含む。				
8.1	【属性】 label	地物属性値を一意に識別する記述ラベル。	M	1	CharacterString	
8.2	【属性】 code	地物属性値を一意に識別する英数字のコード。	0	1	CharacterString	



8.3	【属性】 definition	自然言語で記述される属性値の定義。	0	1	CharacterString	
8.4	【関連役割】 definitionReference	定義の情報源への関連。	0	1	FC_DefinitionReference	
9	【クラス】 FC_FeatureAssociation	地物型を同一又は異なる地物型と結び付ける関連。				
9.1	【属性】 typeName	地物カタログ内で地物関連を一意に識別するための文字列。	M	1	CharacterString	
9.2	【属性】 definition	自然言語で記述する地物関連の定義。	0	1	CharacterString	
9.3	【属性】 alias	地物関連の名前と同等の名前。	0	N	CharacterString	
9.4	【関連役割】 featureCatalogue	地物関連が含まれる地物カタログへの関連。	M	1	FC_FeatureCatalogue	
9.5	【関連役割】 definitionReference	地物関連の定義情報源への関連。	0	1	FC_DefinitionReference	
9.6	【関連役割】 role	関連の部分となる関連役割。	M	N	FC_AssociationRole	合成
10	【クラス】 FC_DefinitionSource	定義の情報源を特定する型。				
10.1	【属性】 source	文書や取得方法が識別できるに十分な実際の情報源への引用。	M	1	CI_Citation	
11	【クラス】 FC_DefinitionReference	データインスタンスを定義の情報源に関連付ける型。				
11.1	【属性】 sourceIdentifier	情報源において定義を配置することを支援するための付加的な情報。	0	1	CharacterString	
11.2	【関連役割】 definitionSource	定義参照を情報源の引用情報に関連付ける役割。	M	1	FC_DefinitionSource	

	名称	コード	記述
12	FC_RoleType		関連役割の種別のための列挙値型。
12.1	ordinary	1	通常の関連。
12.2	aggregation	2	集成。
12.3	composition	3	合成。

## 12 符号化

**序文** このプロファイルは、ISO 19118 (Geographic information – Encoding) から、符号化規則を作成するために必要となる基本的な規則を抽出及び翻訳し、プロファイルとして定義したものである。

### 12.1 基本的な概念

#### 12.1.1 一般

このプロファイルは、さまざまな地理情報システム間における空間データの相互利用を可能にすることを目的とする。そのためには、以下の二つの基本的な課題を解決する必要がある。一つ目の課題は、空間データ的内容及び論理構造の意味を定義することである。これは、「応用スキーマ」として規定する。二つ目の課題は、その応用スキーマに対応するデータを表現できる、システムやプラットフォームに依存しないデータ構造を定義することである。

ここでは、データ交換の基本概念、すなわち、応用スキーマに基づいて空間データを符号化、送付、受領、及び解釈するための手続きのうち、データ交換過程の概要 (12.1.2)、システムに依存しないデータ構造を生成するための符号化規則 (12.3) について規定する。

**備考** このプロファイルに準拠した符号化規則の一つとして、「附属書 8 (参考) XML に基づく符号化規則」を示す。またここでは、XML に基づく符号化規則に従い、このプロファイルで取り上げる基本データ型、空間スキーマ、時間スキーマ、被覆の幾何及び関数のためのスキーマ、地理識別子による空間参照、参照系、他の規格から引用するクラスの W3C XML Schema 文書を示す。

#### 12.1.2 データ交換

データ交換の概念を図 12-1 に示す。

システム A からシステム B にデータ集合を送るとする。A と B の良好なデータ交換を保証するためには、次の三つについて取り決めなければならない。すなわち、共通の応用スキーマ I、適用する符号化規則 R、そして使用する転送プロトコルの種類である。応用スキーマは、良好なデータ転送を実現するための基本となるものであり、転送されるデータのあり得べき内容と構造を定義するものである。また、符号化規則はデータをシステムに依存しないデータ構造に変換するための規則を定義する。

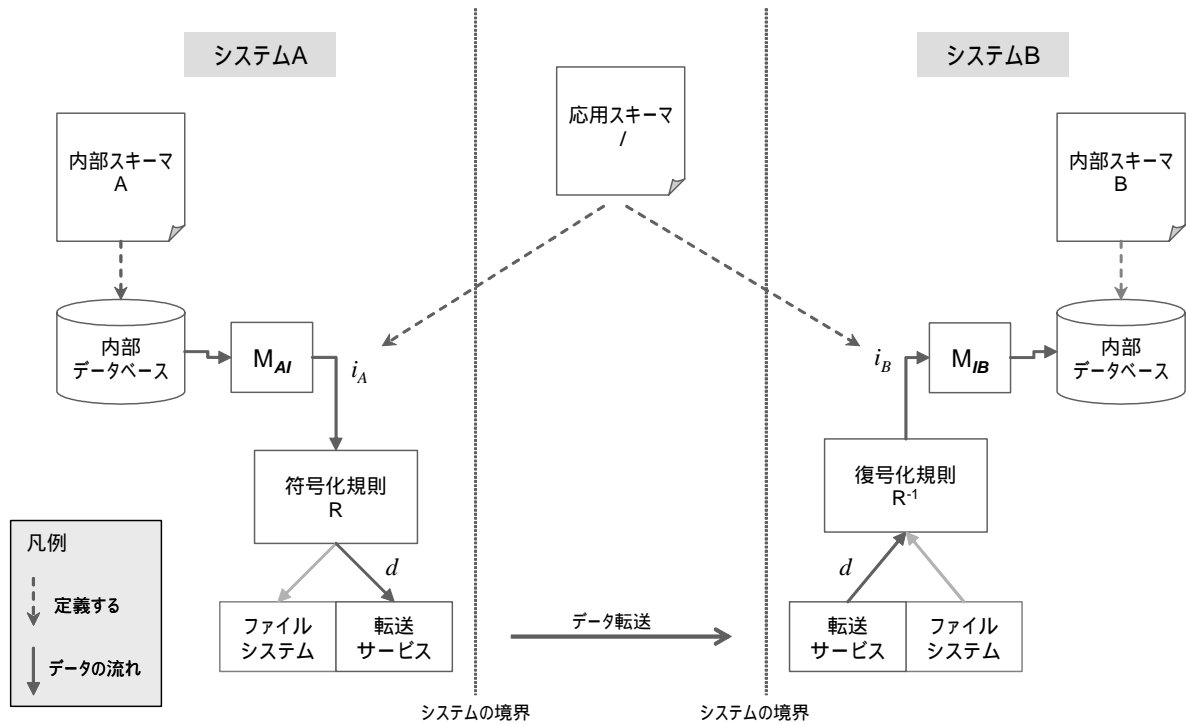


図12-1 二つのシステム間におけるデータ交換の概念

A, B 両システムとも、データを内部スキーマに従って内部データベースに保持しているが、通常、スキーマは異なる。すなわち、スキーマ  $A \neq$  スキーマ B。A の内部データベースから B の内部データベースへデータ集合を転送するためには、次の論理的ステップを踏まなければならない。このプロファイルは、符号化規則を作る際の要件を規定するものであり、データ交換プロセス全体は規定しない。従って、ステップ b) 及び e) のみを標準化する。ステップ c) 及び d) は、一般的な情報技術の転送サービスを使用することになる。

- まず、システム A において、その内部データを共通の応用スキーマ I に従ったデータ構造に変換する。これは、内部スキーマの概念から応用スキーマにおいて定義されている概念への写像を定義し、個々のデータ・インスタンスを変換する適切な対応付けソフトウェアを実装することにより行われる。図 12-1において、この写像は  $M_{AI}$  で表している。 $M_{AI}$  の出力は、応用スキーマ特有データ構造  $i_A$  である。このデータ構造は、メモリ又は中間ファイルに記憶されるが、システム A に依存したものであり、そのまま相手にデータ転送するには適していない。
- 次に、符号化サービスを使う。これは、システムに依存しない転送に適しているデータ構造を作成するために、符号化規則 R を適用する。この符号化されたデータ集合 d は、ファイルシステムに保管してもよいし、転送サービスを使用して転送してもよい。
- システム A は、次に、符号化されたデータ集合 d をシステム B に送るために転送サービスを呼び出す。転送サービスは、パッケージの仕方やオンライン又はオフラインの通信媒体上の実際の伝送方式に関して、転送プロトコルに従う。両者は、使用する転送プロトコルについて合意していなければならない。
- システム B の転送サービスは、転送されたデータ集合を受け取り、転送プロトコルに従ってデータ集合をアンパックし、符号化されたデータ集合 d として、例えば中間ファイルに保管する。

- e) システム B は、応用スキーマ特有データ構造のデータ  $i_B$  を得るために、符号化されたデータを解釈する復号化規則  $R^{-1}$  を適用する。
- f) システム B では、転送されたデータ集合を使用するために、応用スキーマ特有データ構造のデータ  $i_B$  を内部データベースに変換しなければならない。この変換は、応用スキーマから内部スキーマへの写像を定義し、実際の変換を行うソフトウェアを実装することによって行われる。図 12-1において、この写像は  $M_{IB}$  で表している。

## 12.2 文字レパートリ

このプロファイルは、国際文字の規格で定義されている JIS X 0221 を採用する。これらの規格は国際文字集合 (UCS) と呼ばれる国際的に認められた文字レパートリとその文字符号化体系を定義する。

このプロファイルの国際プロファイルにもとづく文字符号化体系は、以下のとおりである。

- a) 8 ビット可変長 UCS 転送フォーマット UTF-8
- b) 16 ビット可変長 UCS 転送フォーマット UTF-16
- c) 16 ビット固定長国際文字集合 UCS-2
- d) 32 ビット固定長国際文字集合 UCS-4

上記に加え、このプロファイルでは、日本において利用性が高いと考えられる以下の三つを文字レパートリとして定義する。

- e) Shift\_JIS
- f) EUC-JP
- g) Windows-31J

## 12.3 符号化規則

### 12.3.1 概説

ここでは、符号化規則の規定のための要件を規定する。このプロファイルに準拠した応用スキーマを作成する場合は、符号化規則を規定しなければならない、応用スキーマと合わせて使用者には提供しなければならない。

符号化規則は一般に以下のことを規定しなければならない。

- a) 符号化要件概要 (12.3.2)
  - 1) 応用スキーマとスキーマ言語 (12.3.2.1)
  - 2) サポートしている国際文字レパートリと言語 (12.3.2.2)
  - 3) データ集合とオブジェクト識別スキーマ (12.3.2.3)
- b) 入力データ構造 (12.3.3)
  - 1) 応用スキーマに基づいてデータを保管するためのデータ構造としてのインスタンスモデル
  - 2) インスタンスモデルを応用スキーマに関連付ける方法
- c) 交換フォーマットとなる出力データ構造 (12.3.4)
- d) インスタンスモデルのデータを交換フォーマットに変換するための変換規則 (12.3.5)。マッピング規則という。
  - 1) 符号化のための変換規則

### 12.3.2 符号化要件概要

#### 12.3.2.1 応用スキーマとスキーマ言語

符号化規則は応用スキーマを定義するために使用するスキーマ言語を規定し、かつ、どのように応用スキーマが構成されるかを記述しなければならない。

#### 12.3.2.2 文字レパートリと言語

文字レパートリは使用可能な文字を定義する。符号化規則は、使用する文字レパートリ及びサポートする言語と言語を特定する仕組みを規定しなければならない。

### 12.3.2.3 データ集合

#### 12.3.2.3.1 データ集合の要素と構造

効率的に符号化、復号化するためには、データ集合の要素と構造を定義することが重要である。オブジェクトが情報の基本単位と考えられ、異なる型のオブジェクトは、他の型のオブジェクトと識別できなければならない。データ集合の中のオブジェクトは、列にする又は階層構造にすることができる。オブジェクトは属性の列として内部的に構造化してもよいし、また他のオブジェクトへの参照を含んでもよい。階層的な手法によって他のオブジェクトで構成してもよい。

符号化規則は以下の内容を規定しなければならない。

- a) オブジェクトが何であるか、及び、オブジェクトの型
- b) オブジェクトの構造
- c) データ集合の構造

#### 12.3.2.3.2 オブジェクト識別

オブジェクトは、特定の状況の中で一意に識別できる識別子をもつことができる。二つの異なる状況を考慮しなければならない。

一つ目の状況はデータ集合である。オブジェクト識別子 (ID) は特定のデータ集合の中で一意である。これらの識別子はオブジェクトがデータ集合の中でオブジェクトを参照することを可能とする。オブジェクト識別子は通常、オブジェクトがデータ集合に挿入されるときに割り当てられ、本来的には一時的なものである。

二つ目の状況はアプリケーション定義域である。アプリケーション定義域は、分野とユニバーサル一意識別子 (UUID) と呼ばれる識別の仕組みを定義する。UUID は、オブジェクトが生成されるときに割り振られ、オブジェクトが存在している限り不変である。削除されたオブジェクトの UUID は、再び使用することはできない。UUID は、長期の分散データ管理と更新の仕組みを実現するために必要とされる。これらの識別子は、また、永続的識別子と呼ばれる。UUID は、アプリケーション定義域によって明確に定義され、かつ限定された領域の中で一意である。

符号化規則は、以下の内容を規定しなければならない。

- a) 使用する、異なるオブジェクトの識別メカニズム
- b) 識別メカニズムの内部構造

### 12.3.3 入力データ構造

入力データ構造はインスタンスモデルと呼ばれ、アプリケーションデータについての意味付け、変換規則の定義付けを表現するための道具である。インスタンスモデルは、応用スキーマの仕様に従ってデータを表すことができなければならない。それは、特定の応用スキーマに特有のものであってもよいし、又はどのようなスキーマに従ってでも表現できるような一般的なものでよい。

符号化規則は、インスタンスモデルとその応用スキーマへの関係を明示しなければならない。

### 12.3.4 出力データ構造

出力データ構造は、どのようにデータが構造化され、交換ファイルの中で表現されるかを定義する。スキーマが出力データ構造に伴ってもよい。

符号化規則は、出力データ構造を明示しなければならない、存在する場合には、出力データ構造スキーマを明示しなければならない。

### 12.3.5 変換規則

変換規則は、入力データ構造の中のデータインスタンスが、出力データ構造のインスタンスにどのように変換されるかを明示する。二つの変換規則の対が存在することができる。一つ目はスキーマ変換規則で、UML スキーマから出力データ構造のスキーマへの写像を定義する。二つ目はインスタンス変換規則で、インスタンスモデルのインスタンスから結果のデータ構造のインスタンスへの写像を定義する。（図 12-2）

符号化規則は、以下の内容を規定しなければならない。

- a) スキーマ変換規則
- b) インスタンス変換規則

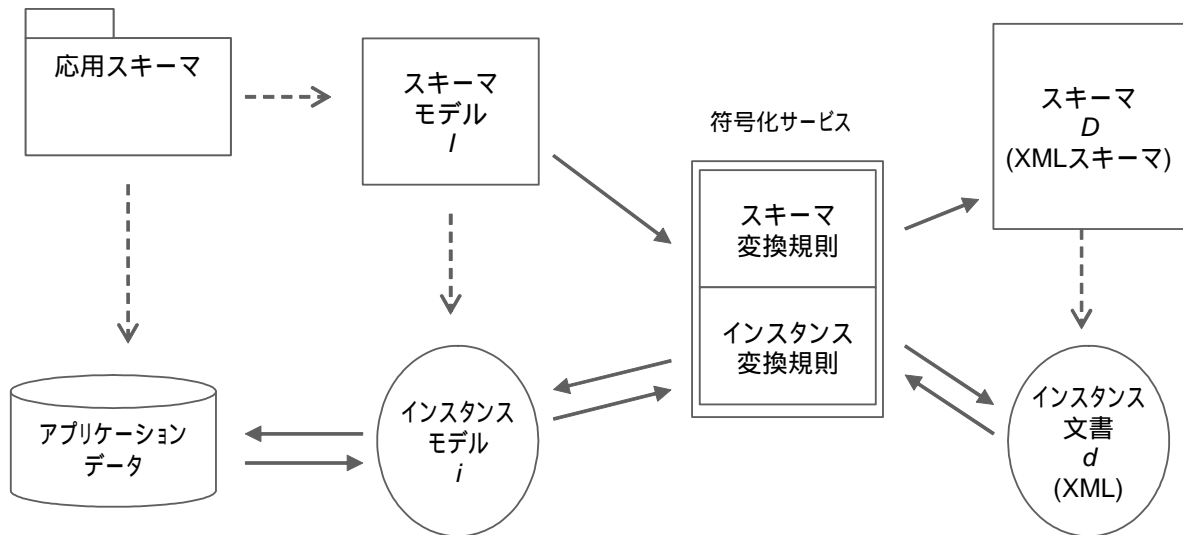


図12-2 変換規則

### 12.3.6 附属書について

このプロファイルでは、符号化規則は定めないが、附属書 8 又は附属書 12 に示す符号化規則を使用することができる。

## 附属書 1 (規定) 抽象試験項目群

### 1. 適合性の枠組み

#### 1.1 抽象試験項目群

抽象試験項目群は、空間データ製品仕様書及び応用システムが JPGIS に適合するか検査する、特定の要件のために作成された試験の集まりである。ISO 19100 シリーズ及び JIS X 7100 シリーズは、適合性要件を含まなければならない。JPGIS においても次に示す項目について抽象試験項目群を示し、この附属書で規定する。JPGIS への適合を主張する場合には、必ず抽象試験項目群に基づいた検査を行わなければならない。

- 「6. 応用スキーマのための規則」
- 「7. 空間スキーマ」
- 「8. 時間スキーマ」
- 「9. 被覆の幾何及び関数のためのスキーマ」
- 「10. 地理識別子による空間参照」
- 「11. 地物カタログ化法」
- 「12. 符号化」
- 「附属書 2. (規定) 参照系」
- 「附属書 6. (規定) 他の規格から引用するクラスの定義」

ところで、実行可能試験項目群とは、特定の要件を満たす試験対象に固有の試験のことであり、抽象試験項目に値を入れ、具体化すると、実行可能試験項目ができる。実行可能試験項目群は試験対象に固有の試験のため、このプロファイルの中で示すことはできない。したがって、製品仕様書及び応用システムを検査しようとする者は、それぞれの試験対象に応じて実行可能試験項目群を作成する。

抽象試験項目群は、JIS X 7105 (地理情報 — 適合性及び試験) で規定する、適合性を評価するための「適合性評価過程」で使用する。適合性評価過程は、次の四つの段階からなる。

- ① 試験の準備
- ② 試験の実施
- ③ 結果の解析
- ④ 適合性試験の報告

抽象試験項目群は、この中の「① 試験の準備」において、実行可能試験項目群は「② 試験の実施」において使用する。

#### 1.2 基本試験と機能試験

基本試験とは、試験対象について全体を通して試験を行うのが適切であるかどうかを判断するために限定的に行う試験である。この試験によって、試験対象が抽象試験項目に適合していることの予備的な証拠を得ることができる。

一方、機能試験とは、「実装適合性宣言」に記述された機能に合致しているかどうかを調べる試験であり、規定した適合性要件のすべての範囲について、試験対象を実際に可能な限り全体を通して調べなければならない。実装適合性宣言とは、仕様の中での任意に選択してよいとされる要件について、そのうちどれを実装したのかを記述した文書である。したがって、機能試験を行うには、「実装適合性宣言」の作成が必要である。

しかし、このプロファイルの中の任意の選択要件から何を実装したかは応用スキーマを見れば明らかことが多い。そのような場合は、実装適合性宣言は作成しなくてよい。

また、ISO 19100 シリーズ及び JIS X 7100 シリーズでは試験の種類を「基本試験」としている抽象試験項目も存在するが、このプロファイルでは、試験の種類はすべて「機能試験」とする。これは、限定的な範囲から得られる予備的な証拠では、このプロファイルへの適合性を完全に判断することができないからである。

抽象試験項目は、次の四つの項目を含まなければならない。

- a) 試験目的・・・試験を行う意図を正確に示す。
- b) 試験方法・・・試験の手順を示す。
- c) 参照・・・関連する特定の箇条を示す。
- d) 試験種類・・・基本試験又は機能試験のいずれかを示す。

### 1.3 JPGIS への適合性

JPGIS では、適合性を、「適合」「不適合」「部分適合」の三つに分類する。

空間データ製品仕様書及び応用システムが、すべての適合性試験に合格する場合、それらは「JPGIS に完全に準拠している」と主張することができる。

空間データ製品仕様書及び応用システムが、すべての適合性試験に合格しない場合、それらは「JPGIS に準拠している」と主張してはならない。

空間データ製品仕様書及び応用システムが、一部の適合性試験に合格する場合、どの抽象試験項目に適合するのかを明示しなければならない。これを部分適合という。

## 2. 応用スキーマのための規則

### 2.1 地物の定義

#### 2.1.1 地物型の定義

- a) **試験目的** 地物型が一般地物モデルに従って定義されていることを確認する。
- b) **試験方法** 地物型が附属書 1 の 2.1.1 c) に示す要素と規則に従って定義されていることを検査する。
- c) **参照** 6.1.3
- d) **試験種類** 機能

#### 2.1.2 地物属性型の定義

- a) **試験目的** 地物属性型が一般地物モデルに従って定義されていることを確認する。
- b) **試験方法** 地物属性型が附属書 1 の 2.1.2 c) に規定される要素と規則に従って定義されていることを検査する。
- c) **参照** 6.1.5 及び 6.2
- d) **試験種類** 機能

#### 2.1.3 地物関連型の定義

- a) **試験目的** 地物関連型が一般地物モデルに従って定義されていることを確認する。
- b) **試験方法** 地物関連型が附属書 1 の 2.1.3 c) に規定される要素と規則に従って定義されていることを検査する。
- c) **参照** 6.1.8
- d) **試験種類** 機能



## 2.2 UML による応用スキーマの作成

### 2.2.1 応用スキーマの識別と統合

- a) **試験目的** このプロファイルに規定された規則に従った、他の ISO に関連した概念スキーマで、応用スキーマが識別され統合されていることを確認する。
- b) **試験方法** 附属書 1 の 2.2.1 c) に規定される規則が応用スキーマの識別と統合に従って定義されていることを検査する。
- c) **参照** 6.3.1~6.3.3
- d) **試験種類** 機能

### 2.2.2 応用スキーマにおける地物型の表現

- a) **試験目的** 地物型が応用スキーマにおいてこのプロファイルで規定される規則に従って表現されていることを確認する。
- b) **試験方法** 地物の表現が附属書 1 の 2.2.2 c) の一般規則に従っていることを検査する。
- c) **参照** 6.3.4
- d) **試験種類** 機能

### 2.2.3 応用スキーマにおける空間スキーマ要素の使用

- a) **試験目的** 空間スキーマの要素が応用スキーマで表現される場合、このプロファイルに規定された規則に従って表現されているかを確認する。
- b) **試験方法** 空間スキーマの要素の表現が附属書 1 の 2.2.3 c) の規則に従っていることを検査する。
- c) **参照** 6.4
- d) **試験種類** 機能

### 2.2.4 応用スキーマにおける時間スキーマ要素の使用

- a) **試験目的** 時間スキーマ要素が応用スキーマで表現される場合、このプロファイルに規定された規則に従って表現されているかを確認する。
- b) **試験方法** 時間スキーマ要素の表現が附属書 1 の 2.2.4 c) の規則に従っていることを検査する。
- c) **参照** 6.5
- d) **試験種類** 機能

### 2.2.5 応用スキーマにおける被覆スキーマ要素の使用

- a) **試験目的** 被覆スキーマが応用スキーマの中で表現される場合、このプロファイルに規定された規則に従って表現されているかを確認する。
- b) **試験方法** 被覆スキーマ要素の表現が附属書 1 の 2.2.5 c) の規則に従っていることを検査する。
- c) **参照** 6.7
- d) **試験種類** 機能

### 2.2.6 応用スキーマにおける地名辞典スキーマの地理識別子の使用

- a) **試験目的** 地理識別子による参照が応用スキーマで表現される場合、このプロファイルに規定された規則に従って表現されているかを確認する。
- b) **試験方法** 地名辞典スキーマの表現が附属書 1 の 2.2.6 c) の規則に従っていることを検査する。

- c) 参照 6.8
- d) 試験種類 機能

### 2.2.7 応用スキーマにおける地物カタログスキーマの使用

- a) 試験目的 地物カタログスキーマからの情報が応用スキーマに表現される場合、このプロファイルに規定された規則に従って表現されているかを確認する。
- b) 試験方法 地物カタログに定義された地物の表現が附属書 1 の 2.2.7 c) の規則に従っていることを検査する。
- c) 参照 6.9
- d) 試験種類 機能

## 3. 空間スキーマ

### 3.1 序文

このプロファイルでは、抽象試験項目群を、二つの基準によって八つの項目に分類する。最初の基準はデータの複雑さによる基準であり、四つの分類項目を定める。

- ・ 幾何プリミティブ
- ・ 幾何複体
- ・ 位相複体
- ・ 幾何実現をもつ位相複体

**備考** 一般に“スパゲッティ”データと呼ばれるデータのスキーマは、幾何プリミティブの構造化されていない集まりのみを用いる。幾何のそれぞれの構成要素について、それぞれ単一の定義が必要な場合、応用スキーマには幾何複体を使用される。同一の幾何複体内のプリミティブは境界のみを共有する。このスキーマに明示的な位相情報が必要な場合には、幾何複体に位相複体の構造を関連させる。複体に含まれるオブジェクトの型は複体の次元に支配される。

第二の基準は次元による基準であり、三つの分類項目を定める。

- ・ 零次元オブジェクト
- ・ 零及び一次元オブジェクト
- ・ 零、一及び二次元オブジェクト

ただし、零次元複体は零次元幾何プリミティブの集合以上に有益な情報を提供しないため、複体については一次元及び二次元のみとする。

また、このプロファイルでは、二次元幾何複体 (GM\_CompositeSurface) を含まないため、幾何複体に関しては一次元のみとする。

### 3.2 幾何プリミティブ

#### 3.2.1 零次元幾何プリミティブのデータ型

- a) 試験目的 応用スキーマが position 属性をもつ GM\_Point をインスタンス化することを確認する。それが CoordinateReferenceSystem 関連をもつ場合、その表現が規定通りにされているか確認する。
- b) 試験方法 応用スキーマの文書を検査する。
- c) 参照 7.1, 7.2.1, 7.2.2.5, 7.2.2.6, 7.2.3.1 及び 7.5
- d) 試験種類 機能

### 3.2.2 一次元幾何プリミティブのデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.2.1 の要件を満たすと同時に、orientation 属性及び Segmentation 関連をもつ GM\_Curve 並びにすべての属性をもつ GM\_CurveSegment の一つ以上の下位型をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.2.1, 7.2.2.7, 7.2.2.8, 7.2.2.10, 7.2.3.1 及び 7.2.3.4~7.2.3.9
- d) **試験種類** 機能

### 3.2.3 二次元幾何プリミティブのデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.2.2 のすべての要件を満たし、orientation 属性及び Segmentation 関連をもつ GM\_Surface 並びにすべての属性をもつ GM\_SurfacePatch の一つ以上の下位型をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.2.2, 7.2.2.9, 7.2.2.11, 7.2.3.4 及び 7.2.3.10~7.2.3.15
- d) **試験種類** 機能

## 3.3 幾何複体

### 3.3.1 一次元幾何複体のデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.2.2 のすべての要素を満たすと同時に、GM\_Complex 又は GM\_CompositeCurve をインスタンス化することを確認する。それが、各 GM\_Primitive (GM\_Point 及び GM\_Curve) と GM\_Complex との間の Complex 関連又は GM\_OrientableCurve と GM\_CompositeCurve との間の Composition 関連をもつことを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.2.2, 7.2.4.1 及び 7.2.4.2
- d) **試験種類** 機能

## 3.4 位相複体

### 3.4.1 一次元位相複体のデータ型

- a) **試験目的** 応用スキーマが TP\_Node, TP\_DirectedNode, TP\_Edge 及び TP\_DirectedEdge をインスタンス化し、TP\_DirectedNode 及び TP\_DirectedEdge をインスタンス化したものが共に orientation 属性をもつことを検証する。応用スキーマが TP\_Complex とそれぞれの TP\_Primitive (TP\_Node 及び TP\_Edge) との間の Complex 関連をもつことを検証する。さらに、TP\_Node と TP\_DirectedNode との間及び TP\_Edge と TP\_DirectedEdge との間の Center 関連をもつことを確認する。これが TP\_DirectedNode と TP\_Edge との間の Boundary 関連をもつことを検証する。また、TP\_Node と TP\_DirectedEdge との間の CoBoundary 関連がある場合、その表現が規定通りされているか確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 7.3.1, 7.3.3.1~7.3.3.6 及び 7.3.4.1
- d) **試験種類** 機能

### 3.4.2 二次元位相複体のデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.4.1 のすべての要件を満たすと同時に TP\_Face 及び TP\_DirectedEdge をインスタンス化することを検証する。TP\_DirectedFace をインスタンス化したものが orientation 属性に対応することを検証する。応用スキーマが TP\_Complex と TP\_Face との間の Complex 関連及び TP\_Face と TP\_DirectedFace との間の Center 関連に対応

することを検証する。これが TP\_DirectedEdge と TP\_Face との間の Boundary 関連をもつことを検証する。また、TP\_Edge と TP\_DirectedFace との間に CoBoundary 関連がある場合、その表現が規定通りされているか確認する。

- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.4.1, 7.3.3.7 及び 7.3.3.8
- d) **試験種類** 機能

### 3.5 幾何実現を伴う位相スキーマ

#### 3.5.1 幾何実現を伴う一次元位相スキーマのデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.2.2 及び附属書 1 の 3.4.1 のすべての要件を満たすことを確認する。それがまた、TP\_Primitive (TP\_Node 及び TP\_Edge) と GM\_Primitive (GM\_Point 及び GM\_Curve) とをそれぞれインスタンス化したもの間、及び TP\_Complex と GM\_Complex とをそれぞれインスタンス化したもの間に Realization 関連がある場合、その表現が規定通りにされているか確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.2.2, 附属書 1 の 3.4.1 及び 7.4
- d) **試験種類** 機能

#### 3.5.2 幾何実現を伴う二次元位相スキーマのデータ型

- a) **試験目的** 応用スキーマが附属書 1 の 3.2.3 及び附属書 1 の 3.4.2 のすべての要件を満たすと同時に、TP\_Face と GM\_Surface とをそれぞれインスタンス化したもの間、及び TP\_Complex と GM\_Complex とをそれぞれインスタンス化したもの間の Realization 関連がある場合、その表現が規定通りにされているか確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 3.2.3, 附属書 1 の 3.4.2 及び 7.4
- d) **試験種類** 機能

## 4. 時間スキーマ

### 4.1 幾何スキーマ

- a) **試験目的** 応用スキーマが position 属性をもつ TM\_Instant, 又は Beginning 及び Ending 関連をもつ TM\_Period をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 8.2.2, 8.2.3 及び 8.5.3
- d) **試験種類** 機能

### 4.2 位相スキーマ

- a) **試験目的** 応用スキーマが TM\_Node, 又は Initiation 及び Termination 関連をもつ TM\_Edge をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 8.3.3 及び 8.3.4
- d) **試験種類** 機能

### 4.3 幾何実現

- a) **試験目的** 応用スキーマが附属書 1 の 4.1 及び附属書 1 の 4.2 の要件を満たすと同時に、TM\_Node と TM\_Instant との間の Realization 関連及び TM\_Edge と TM\_Period との間に Realization 関連がある場合、その表現が規定通りにされているか確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 4.1, 附属書 1 の 4.2, 8.4.2 及び 8.4.3
- d) **試験種類** 機能

## 5. 被覆の幾何及び関数のためのスキーマ

### 5.1 序文

この箇条では、二種類の抽象試験項目を示す。一種類目には、各種被覆型の評価に対応する単純なインタフェースを利用可能にする共通の抽象試験項目が含まれるが、被覆の種類までは触れない。二種類目には、個別の被覆型に対応するインタフェースを利用可能とし、被覆の内部構造に関する追加的な情報を明らかにするような抽象試験項目が含まれている。

### 5.2 単純被覆インタフェース

- a) **試験目的** 応用スキーマが、domainExtent 属性及び CRS 関連役割をもつ CV\_Coverage をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 9.2.1
- d) **試験種類** 機能

### 5.3 離散被覆インタフェース

- a) **試験目的** 応用スキーマが、附属書 1 の 2.2.5 の要件を満たし、CoverageFunction 関連をもつ CV\_DiscreteCoverage とその下位型をインスタンス化し、インスタンス化した型が CV\_DiscretePointCoverage, CV\_DiscreteCurveCoverage, 又は CV\_DiscreteSurfaceCoverage の場合は、geometry 及び value 属性をもつ CV\_GeometryValuePair とその下位型をインスタンス化することを確認する。インスタンス化した型が CV\_DiscreteGridPointCoverage の場合は、CV\_GridValuesMatrix をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 2.2.5, 9.2.2~9.2.5 及び 9.3
- d) **試験種類** 機能

### 5.4 ティーセン多角形被覆

- a) **試験目的** 応用スキーマが、附属書 1 の 2.2.5 の要件を満たし、規定された属性及び関連によって、CV\_ThiessenPolygonCoverage 及び CV\_ThiessenValuePolygon をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 1 の 2.2.5 及び 9.4
- d) **試験種類** 機能

### 5.5 不規則三角網被覆

- a) **試験目的** 応用スキーマが、附属書 1 の 2.2.5 の要件を満たし、規定された属性及び関連によって、CV\_TINCoverage 及び CV\_ValueTriangle をインスタンス化することを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。

- c) **参照** 附属書1の2.2.5及び9.5
- d) **試験種類** 機能

## 6. 地理識別子による空間参照

### 6.1 場所型

- a) **試験目的** 各場所型が明確になっており、一つ又は複数の場所インスタンスによって一意に識別されていることを検査する。
- b) **試験方法** 各場所型の属性が識別されており、場所インスタンスの地名辞典が存在することを検査する。
- c) **参照** 10.2
- d) **試験種類** 機能

### 6.2 地名辞典

- a) **試験目的** 地名辞典の構造を検査する。
- b) **試験方法** 地名辞典のプロパティが識別されていることを検査する。
- c) **参照** 10.2.2
- d) **試験種類** 機能

### 6.3 場所インスタンス

- a) **試験目的** 地名辞典に記録される各場所インスタンスのすべての属性が、正しく定義されていることを検査する。
- b) **試験方法** 地名辞典に記録される場所のすべてのインスタンス及びそれぞれの属性データが、正しく記録されていることを検査する。
- c) **参照** 10.2.3
- d) **試験種類** 機能

## 7. 地物カタログ化法

### 7.1 地物カタログの形式のための試験項目

- a) **試験目的** 地物カタログの形式を検査する。
- b) **試験方法** 地物カタログがデジタル形式で作成され、コンピュータのディスク装置やファイル転送による地物カタログのコピーの取得が可能であることを確認する。
- c) **参照** 11.2.1
- d) **試験種類** 機能

### 7.2 地物カタログの一般要件のための試験項目

- a) **試験目的** 地物カタログの一般要件を満たしていることを検査する。
- b) **試験方法** 次の項目を確認する。
  - 1) 対象となるすべての地物型について地物カタログが文書化されている。
  - 2) すべての地物型、地物属性、地物関連及び関連役割が名前によって地物カタログ内で一意に識別される。
  - 3) すべての地物属性定義域が、列挙型地物属性の中で名前によって一意に識別される。
  - 4) すべての地物型、地物属性、地物関連及び関連役割が定義属性で定義しているか、又は他の定義情報源を参照している。

- c) **参照** 11.2.2
- d) **試験種類** 機能

### 7.3 地物カタログクラスの試験項目

- a) **試験目的** 地物カタログが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.1 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.1
- d) **試験種類** 機能

### 7.4 地物型の試験項目

- a) **試験目的** 地物型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.2 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.2
- d) **試験種類** 機能

### 7.5 継承関連型の試験項目

- a) **試験目的** 継承関連型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.3 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.3
- d) **試験種類** 機能

### 7.6 結合型の試験項目

- a) **試験目的** 結合型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.5 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.5
- d) **試験種類** 機能

### 7.7 地物属性型の試験項目

- a) **試験目的** 地物属性型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.6 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.6
- d) **試験種類** 機能

### 7.8 関連役割型の試験項目

- a) **試験目的** 関連役割型のオブジェクトが必須情報を含んでいることを検査する。

- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.7 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.7
- d) **試験種類** 機能

#### 7.9 列挙値型の試験項目

- a) **試験目的** 列挙値型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.8 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.8
- d) **試験種類** 機能

#### 7.10 地物関連型の試験項目

- a) **試験目的** 地物関連型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.9 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.9
- d) **試験種類** 機能

#### 7.11 定義情報源の試験項目

- a) **試験目的** 定義情報源型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.10 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.10
- d) **試験種類** 機能

#### 7.12 定義参照型の試験項目

- a) **試験目的** 定義参照型のオブジェクトが必須情報を含んでいることを検査する。
- b) **試験方法** 個々の属性及び役割の記述，必須/条件付き，多重度最大値，型及び制約が要件を満たしていることを検査することによって，表 11-1.11 に示した個々の属性及び役割を調べる。
- c) **参照** 表 11-1.11
- d) **試験種類** 機能

### 8. 符号化

#### 8.1 一般符号化要件

- a) **試験目的** 符号化規則が 12.3 に示す符号化規則の要件に合致しているか検証する。
- b) **試験方法** 符号化規則を検査し，すべての要件に合致していることを調査する。
- c) **参照** 12.3
- d) **試験種類** 機能



## 9. 参照系

### 9.1 RS\_CRS

- a) **試験目的** RS\_CRS が、附属書 2 に規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 2 の 2.3
- d) **試験種類** 機能

### 9.2 RS\_LRS

- a) **試験目的** RS\_LRS が、このプロファイルに規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 2 の 2.4
- d) **試験種類** 機能

## 10. 他の規格から引用するクラスの定義

### 10.1 EX\_Extent

- a) **試験目的** EX\_Extent が、附属書 6 に規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 6 の 1
- d) **試験種類** 機能

### 10.2 CI\_Citation

- a) **試験目的** CI\_Citation が、附属書 6 に規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 6 の 3
- d) **試験種類** 機能

### 10.3 CI\_ResponsibleParty

- a) **試験目的** CI\_ResponsibleParty が、附属書 6 に規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 6 の 4
- d) **試験種類** 機能

### 10.4 Record 及び RecrdType

- a) **試験目的** Record 及び RecordType が、附属書 6 に規定した属性及び関連に従ってインスタンス化されていることを確認する。
- b) **試験方法** 応用スキーマの文書を検査する。
- c) **参照** 附属書 6 の 5

d) **試験種類** 機能

## 附属書 2 (規定) 参照系

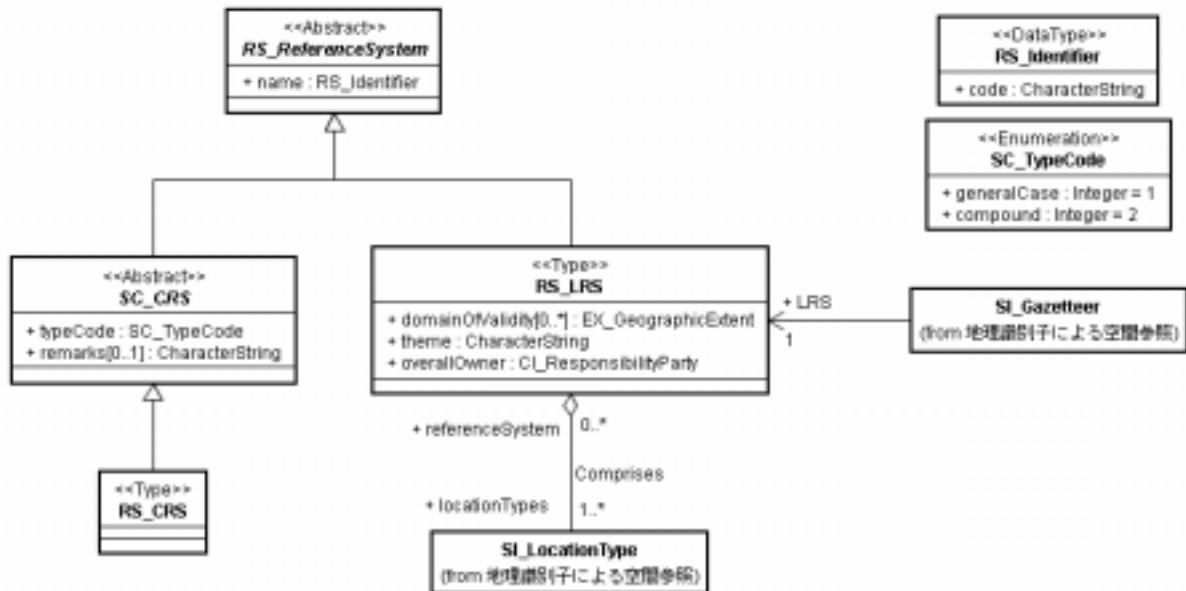
### 1. 参照系の記述

このプロファイルでは、時間参照系及び空間参照系は、この箇条で規定する型を使用する。

また、このプロファイルに基づき作成される空間データ製品仕様書の時間参照系及び座標参照系は、JIS X 7115 の附属書 1 (時間参照系) 及び 2 (座標参照系) に定められた略号の表記規則にしたがって規定する。

### 2. JPGIS 参照系パッケージ

このプロファイルの参照系プロファイルは、JPGIS 参照系パッケージに定義する。



図附 2-1 参照系プロファイル

#### 2.1 RS\_ReferenceSystem

データ集合で使用する参照系についての情報を表す抽象クラス。(JIS X7115 が規定)

##### 属性

1. `name:RS_Identifier` は、使用する参照系の名称を記述する。

#### 2.2 RS\_Identifier

参照系で使用する識別子 (JIS X 7115)。

##### 属性

1. `code:CharacterString` は、使用する参照系の識別子を記述する。このプロファイルでは、時間参照系の場合は JIS X 7115 附属書 1 (時間参照系)、座標参照系の場合は同規格附属書 2 (座標参照系) に定められた略号を記述する。また、地理識別子による空間参照系の場合は、使用する空間参照系の名前を記述する。

### 2.3 SC\_CRS

時間及び座標参照系を表す抽象クラス。RS\_ReferenceSystem の下位型で、一つの属性を継承する。(JIS X7111 が規定)

#### 属性

1. typeCode:SC\_TypeCode は、座標参照系の型コードであり、単一座標参照系のとき 1、複合座標参照系のとき 2 を設定する。
2. remarks[0..1]:CharacterString は、座標参照系に関する注釈を記述する。
3. [上位型から継承する属性]name:RS\_Identifier は、使用する時間及び座標参照系の名称を記述する。

### 2.4 RS\_CRS

このプロファイルで独自に規定する時間及び座標参照系。SC\_CRS の下位型で、三つの属性を継承する。

#### 属性

1. [上位型から継承する属性]name:RS\_Identifier は、使用する時間及び座標参照系の名称を記述する。
2. [上位型から継承する属性]typeCode:SC\_TypeCode は、座標参照系の型コードであり、単一座標参照系のとき 1、複合座標参照系のとき 2 を設定する。
3. [上位型から継承する属性]remarks[0..1]:CharacterString は、座標参照系に関する注釈を記述する。

### 2.5 RS\_LRS

このプロファイルで独自に規定する地理識別子による空間参照系。RS\_ReferenceSystem の下位型で、一つの属性を継承し、三つの属性を独自にもつ。

#### 属性

1. domainOfValidity[0..\*]:EX\_GeographicExtent は、附属書 6 で定義する EX\_GeographicExtent を型として、空間参照系を適用する範囲を示す。
2. theme:CharacterString は、空間参照系の特徴を記述する。
3. overallOwner:CI\_ResponsibleParty は、附属書 6 で定義する CI\_ResponsibleParty を型として、空間参照系に責任をもつものを示す。
4. [上位型から継承する属性]name:RS\_Identifier は、使用する空間参照系の名前を記述する。

#### 関連

1. locationTypes 関連役割は、この RS\_LRS を SI\_LocationType に関連付ける。RS\_LRS は一つ以上の SI\_LocationType で構成しなければならない。

### 附属書 3 (規定) 品質

空間データの品質は、JMP2.0 に準拠するメタデータに記述する。ただし、JMP2.0 を拡張する又はこれによらない場合は、JIS X 7115 に準拠しなければならない。品質の要求、評価及び報告の方式については、別に定める「品質の要求、評価及び報告のための規則」に準拠しなければならない。

## 附属書 4 (規定) メタデータ

このプロファイルに基づき作成される空間データ製品仕様書のメタデータは、JMP2.0 を採用する。ただし、JMP2.0 を拡張する又はこれによらない場合は、JIS X 7115 に準拠しなければならない。

## 附属書 5 (規定) 定義

### 1. ISO19101 のための定義

#### 1.1 応用システム (application)

利用者の要求に応えるために行われる操作及び処理。

#### 1.2 応用スキーマ (application schema)

一つ又は複数の応用システムによって要求されるデータのための概念スキーマ。

#### 1.3 地物 (feature)

現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存在できる。地物型又は地物インスタンスはいずれか一方を意味する場合に用いるべきである。

### 2. 応用スキーマのための規則のための定義

#### 2.1 複合地物 (complex feature)

他の複数の地物から構成される地物。

#### 2.2 データ集合 (dataset)

他と識別可能な、データの集まり。

#### 2.3 定義域 (domain)

矛盾なく定義された値の集合。

**備考** 「矛盾なく定義された」とは、集合の中にあるものは定義を満足するもの全て及び集合の外にあるものは定義を満足しないもの全てであるというような、必要かつ十分な定義を意味する。

#### 2.4 地物関連 (feature association)

地物間の関係。

**備考 1** 地物関連は、型又はインスタンスとして現れてよい。地物関連型又は地物関連インスタンスは、一方のみを意味するときに使われる。

**備考 2** 地物関連は地物の集成を含む。

#### 2.5 地物属性 (feature attribute)

地物の特性。

**備考 1** 地物属性は、型又はインスタンスとして現れてよい。地物属性型又は地物属性インスタンスは、一方のみを意味するときに使われる。

**備考 2** 地物属性型は名前、データ型、及びそれに関連する定義域をもっている。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もっている。

#### 2.6 地理データ (geographic data)

地球に関係した場所への暗示的又は明示的な参照を持ったデータ。

**備考** 地理情報もまた、地球に関係した場所に暗示的又は明示的に関連する現象に関する情報のための用語として使われる。

## 2.7 メタデータ (metadata)

データ記述を含むデータ要素に関するデータ、並びにデータの所有者、アクセス経路、アクセス権及びデータの変更度に関するデータ。

## 2.8 モデル (model)

現実をいくつかの側面から解釈した抽象概念。

## 2.9 論議領域 (universe of discourse)

興味を引くすべてのものを含んだ、実世界又は仮想世界の見方。

## 3. 空間スキーマのための定義

### 3.1 境界 (boundary)

実体の範囲を示す集合。

**備考** 境界が最もよく使用されるのは、集合が点の集まり又はこれらの点を表すオブジェクトの集まりであるような幾何に関する記述である。他の分野では、この用語は比喩的に実体とその論議領域の余事象との遷移領域を記述するために使用する。

### 3.2 合成曲線 (composite curve)

列の各曲線が（最初のものを除き）直前の曲線の終点から開始するような曲線の列。

### 3.3 計算幾何 (computational geometry)

幾何演算を実装するための、幾何表現に関する操作及び計算。

**備考** 計算幾何演算は、幾何包含又は交差の試験、凸包又はバッファ領域の計算若しくは幾何オブジェクト間の最短経路探索を含む。

### 3.4 座標 (coordinate)

$n$ 次元空間内の点の位置を示す  $N$  個の数の列の中の一つの数。

**備考** 座標参照系では、数には単位を付与しなければならない。

### 3.5 座標次元 (coordinate dimension)

ある座標系における位置を記述するために必要な計測値又は座標軸の数。

### 3.6 曲線 (curve)

直線の連続な像を表現する一次元の幾何プリミティブ。

**備考** 曲線の境界はその両端の点の集合である。曲線が閉路の場合、二つの端は同一でありその曲線は（位相的に閉じている場合には）境界をもたないとみなす。最初の点を始点、最後のものを終点と呼ぶ。位相幾何学の定理は連結している集合の連続的な像は連結であることを規定している。

### 3.7 曲線分 (curve segment)



同種の内挿及び定義の方法を用いて曲線の連続した部分を表現するために用いる一次元の幾何オブジェクト。

**備考** 単一の曲線分によって表現される幾何集合はある曲線と等しい。

### 3.8 輪体 (cycle)

境界をもたない空間オブジェクト。

**備考** 輪体は境界の構成要素（殻及び輪を参照）を記述するのに用いられる。輪体はそれ自身の上で閉じているため境界をもたないが、有界である（すなわち、範囲は無限ではない）。例えば、円又は球面は境界をもたないが有界である。

### 3.9 直接位置 (direct position)

座標参照系の中で一組の座標によって記述された位置。

### 3.10 有向エッジ (directed edge)

エッジとその方向の一つとの間の関連を表す有向位相オブジェクト。

**備考** 有向エッジは、そのエッジと方向が一致しているものは + の方向を、そうでなければ - の方向をもつ。有向エッジは位相において同一のエッジの右側 (-) と左側 (+) を区別する場合、同一のエッジの始点ノード (-) と終点ノード (+) を区別する場合及び計算位相でこれらの概念を表現する場合に用いる。

### 3.11 有向フェイス (directed face)

フェイスとその方向の一つとの間の関連を表す有向位相オブジェクト。

**備考** 有向フェイスの外部境界を構成する有向エッジの方向は、このベクトルの方向から正（右側）に見える。位相立体の境界を示す有向フェイスの方向はこの位相立体から離れた方向を指す。隣接する立体はそれらが共有する境界に対して別々の方向を用い、隣接するフェイスとそれらが共有するエッジとの間の関連と同じ方法で一貫性をもつ。有向フェイスは、フェイスとエッジの間の空間的な関係を保持する双対境界関係に用いる。

### 3.12 有向ノード (directed node)

ノードとその方向の一つとの間の関連を表す有向位相オブジェクト。

**備考** 有向ノードは、エッジとノードの間の空間的な関係を保持する双対境界関係に用いる。ノードの方向はエッジに従い、終点ノードは “+”，始点ノードは “-” になる。これは “ $result = end - start$ ” というベクトルの概念と一貫性をもつ。

### 3.13 有向位相オブジェクト (directed topological object)

位相プリミティブとその方向の一つとの間の論理的な関連を表す位相オブジェクト。

### 3.14 領域 (domain)

明確に定義された値の集合。

**備考** 領域は演算子及び関数の定義域及び値域を定義することに用いる。

### 3.15 エッジ (edge)

一次元の位相プリミティブ。

**備考** 領域は演算子及び関数の定義域及び値域を定義することに用いる。エッジの幾何実現は曲線である。エッジの境界は位相複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。

### 3.16 終点ノード (end node)

エッジを含む位相複体の有効な幾何実現において、そのエッジを実現する曲線の終点に対応する、このエッジの境界のノード。

### 3.17 終点 (end point)

曲線の最後の点。

### 3.18 外部 (exterior)

その閉包と全体集合との差集合。

**備考** 外部の概念は位相及び幾何複体の両方に適用できる。

### 3.19 フェイス (face)

二次元の位相プリミティブ。

**備考** フェイスの幾何実現は曲面である。フェイスの境界は、境界関係をとおしてフェイスに関連付けられる同じ位相複体における有向エッジの集合である。これらは輪として形成されることがある。

### 3.20 地理情報 (geographic information)

地球に関連した場所と暗示的又は明示的に関連付けられた現象に関する情報。

### 3.21 幾何集成 (geometric aggregate)

内部構造をもたない幾何オブジェクトの集まり。

### 3.22 幾何境界 (geometric boundary)

幾何オブジェクトの範囲を限定する、より低い幾何次元の幾何プリミティブの集合によって表現される境界。

### 3.23 幾何複体 (geometric complex)

互いに素な幾何プリミティブを含み、各々の幾何プリミティブの境界を、自身のもつ、より低い次元の幾何プリミティブの和集合として表すことのできる集合。

**備考** 集合の幾何プリミティブは、複数の幾何プリミティブの内部に存在するような直接位置は無いという意味で互いに素である。集合は境界演算で閉じているが、これは、幾何複体の各要素について、その要素の境界を表す幾何プリミティブのコレクション（これも幾何複体となる）が存在することを意味する。点（幾何の唯一の零次元プリミティブオブジェクトタイプ）の境界は空集合である。

### 3.24 幾何次元 (geometric dimension)

幾何集合のそれぞれ直接位置を、その直接位置を内部に含み  $n$  次元ユークリッド空間  $R_n$  と相似（同型）な部分集合と関連付けることができるような数  $n$  の最大値。

**備考** 曲線は、実際の線の一部の連続写像であるので、1 の幾何次元をもつ。曲面は、全体を  $R^2$  に展開することは出来ないが、それぞれの点位置の周囲では、微小な近傍は  $R^2$  の単

位円の内部と（連続関数によって）共通する事がわかることから，二次元である。この標準では，主要な曲面分（GM\_SurfacePatch のインスタンス）はその内挿機構の定義によって R2 の部分への写像をもつ。

### 3.25 幾何オブジェクト (geometric object)

幾何集合を表す空間オブジェクト。

**備考** 幾何プリミティブ，幾何プリミティブのコレクション又は幾何複体から成る幾何オブジェクトは単一の実体として扱う。幾何オブジェクトは地物又は地物の意味をもつ一部のようなオブジェクトの空間表現に用いてよい。

### 3.26 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，曲面及び立体を含む。

### 3.27 幾何実現 (geometric realization)

二つの複体の境界関係が一致するような位相複体の位相プリミティブに一対一対応する幾何プリミティブをもつ幾何複体。

**備考** この実現では位相プリミティブは対応する幾何プリミティブの内部を表現するものとみなす。合成したものは閉じている。

### 3.28 幾何集合 (geometric set)

直接位置の集合。

**備考** この集合はほとんどの場合無限集合である。

### 3.29 インスタンス (instance)

クラスを実現するオブジェクト。

### 3.30 内部 (interior)

幾何オブジェクト上でその境界上ではないすべての直接位置の集合。

**備考** 位相オブジェクトの内部は，その任意の幾何実現の内部の準同型の像である。これは，位相幾何学の定理から導かれるので，定義には含まない。

### 3.31 ノード (node)

零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。

### 3.32 点 (point)

位置を表現する零次元の幾何プリミティブ。

**備考** 点の境界は空集合である。

### 3.33 輪 (ring)

輪体である単純曲線。

**備考** 輪は二次元及び三次元座標系における曲面の境界要素を記述するために用いる。

### 3.34 列 (sequence)

繰り返しを許した関連する項目 (オブジェクト又は値) の有限で順序をもつ集まり。

**備考** 論理的には、列は〈項目, 相対位置〉の対の集合である。括弧で列を区切り、列の要素をコンマで分ける LISP 構文を用いる。

### 3.35 集合 (set)

繰り返しを許さない関連する項目 (オブジェクト又は値) の、順序をもたない集まり。

### 3.36 単純 (simple)

内部が等方的 (すべての点が同型な近傍をもつ) で、それによってすべての部分で対応する次元のユークリッド座標空間の開部分集合に局所的に同型となる幾何オブジェクトの特性。

**備考** これは内部の直接位置がどんな種類の自己交差もしないことを意味する。

### 3.37 空間オブジェクト (spatial object)

地物の空間特性を表現するのに用いるオブジェクト。

### 3.38 始点ノード (start node)

エッジを含む位相複体の有効な幾何実現において、そのエッジを実現する曲線の始点に対応する、このエッジの境界のノード。

### 3.39 始点 (start point)

曲線の最初の点。

### 3.40 部分複体 (subcomplex)

すべての要素がより大きな複体の一部でもあるような複体。

**備考** 幾何複体及び位相複体の定義が要求するのは境界演算について閉じることのみであることから、特定の値以下の次元の任意のプリミティブの集合は、常に元のより大きい複体の部分複体になる。したがって、任意の完全平面位相複体は部分複体であるエッジノードグラフを含む。

### 3.41 曲面 (surface)

局所的に平面領域の連続な像を表す二次元の幾何プリミティブ。

**備考** 曲面の境界は、曲面の限界の輪郭を描く向きをもつ閉曲線である。

### 3.42 曲面分 (surface patch)

同種の内挿と定義方法を使って曲面の連続した部分を表わす二次元の連結幾何オブジェクト。

### 3.43 位相複体 (topological complex)

境界演算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。

### 3.44 位相オブジェクト (topological object)

連続した変換のもとでも変わらない空間特性を表わす空間オブジェクト。

### 3.45 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。

## 4. 時間スキーマのための定義

### 4.1 暦 (calendar)

一日をその分解能とする時間位置を定義するための基礎となる離散的な時間参照系。

### 4.2 暦年代 (calendar era)

ある特定の事象から数えた、暦の型としての期間の列で構成する、歴の型の一つ。

### 4.3 協定世界時, UTC (Coordinated Universal Time)

国際度量衡局 (International Bureau of Weights and Measures) 並びに国際地球回転観測事業 (International Earth Rotation Service) によって維持されている標準周期及び時報信号との伝達の基礎を形成する時間の尺度 (ITU-R Rec. TF. 686-1 (1997))。

### 4.4 日 (day)

地球の自転周期にほぼ等しい持続時間をもつ期間。

### 4.5 事象 (event)

ある瞬間に起こる動作。

### 4.6 グレゴリオ暦 (Gregorian calendar)

ユリウス暦よりも一年の長さが太陽年により近くなるよう定義するため、1582年に最初に導入された、はん用的な暦 (JIS X 0301 参照)。

**備考** グレゴリオ暦の導入には、ユリウス暦で積み重なってきた誤差を解消するという意味も含まれていた。グレゴリオ暦の暦年は、平年又はうるう年となっており、一年は、12に分割された月の列からなる。

### 4.7 瞬間 (instant)

時間における位置を表現する零次元の幾何プリミティブ。

### 4.8 ユリウス日 (Julian date)

直前の正午からの経過を表す十進小数の日数を伴ったユリウス日数。

### 4.9 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 日のグリニッジ標準時正午を原点として数えた日の数。

### 4.10 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、物体をモデル化し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。

#### 4.11 月 (month)

月の循環周期の持続時間とほぼ等しい期間。

**備考** 1か月の持続時間は整数で表す日数である。1か月の日数は個々の暦の規則で定められている。

#### 4.12 期間 (period)

時間の範囲を表現する一次元の幾何プリミティブ。

**備考** 二つの異なる時間位置が期間の境界となる。

#### 4.13 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、検索を可能にする時間 (Jensen など (1994))。

#### 4.14 有効時間 (valid time)

ある事実が抽象化した現実の中で真となる時間 (Jensen など (1994))。

### 5. 座標による空間参照のための定義

#### 5.1 座標参照系 (coordinate reference system)

原子により地球に関連づけられた座標系。

#### 5.2 座標系 (coordinate system)

点にどのように座標を割り当てるかを規定するための (数学的) 規則の集合。

#### 5.3 時間座標系 (temporal coordinate system)

単一の時間単位の倍数として距離を測定する間隔尺度を基礎とする時間参照系。

#### 5.4 時間位置 (temporal position)

時間参照系に関する位置。

#### 5.5 時間参照系 (temporal reference system)

時間の測定に対応する参照系。

### 6. 被覆の幾何及び関数のためのスキーマのための定義

#### 6.1 被覆 (coverage)

時空間定義域に含まれる全ての直接位置に対応し、一つ又は複数の地物属性値を戻すような関数として機能する地物。

**例** 被覆の例としては、ラスタ像、多角形オーバーレイ、数値標高マトリクスなどがある。

#### 6.2 被覆幾何 (coverage geometry)

座標により記述した被覆の時空間定義域の構成。

#### 6.3 離散被覆 (discrete coverage)

時空間定義域にある一つの幾何オブジェクトに含まれる全ての直接位置に対して同じ地物属性値を戻す被覆。

**備考** 離散被覆の時空間定義域は、幾何オブジェクトの有限集合で構成される。

#### 6.4 関数 (function)

定義域 (ソース, 又は関数の定義域) から引き出した個々の要素を他の定義域 (対象, 共通定義域, 又は値域) のなかにある一意な要素と関連付けるための規則。

#### 6.5 幾何値オブジェクト (geometry value object)

幾何値対の幾何オブジェクト要素がより大きな幾何オブジェクトの要素であるような幾何値対の集合で構成されるオブジェクト。

#### 6.6 幾何値対 (geometry value pair)

幾何オブジェクトと幾何属性値のレコードで構成される順列をもった対。

#### 6.7 グリッド (grid)

各集合の要素が系統だった方法で他の集合の要素と交差するような複数の曲線の集合で構成されるネットワーク。

**備考** 曲線は、空間をグリッドセルに分割する。

#### 6.8 グリッド点 (grid point)

グリッドのなかで二つ以上の曲線が交差する位置にある点。

#### 6.9 点被覆 (point coverage)

点で構成される時空間定義域をもつ被覆。

#### 6.10 点値対 (point value pair)

幾何要素が点である幾何値対。

#### 6.11 値域<被覆> (range<coverage>)

関数により、被覆の時空間定義域の要素と関連づけられた値の集合。

#### 6.12 ラスタ (raster)

通常は、ブラウン管上の表示を構成したり、これに対応する平行な走査線の方形波パターン。

**備考** ラスタは、グリッドの型である。

#### 6.13 時空間定義域<被覆> (spatiotemporal domain<coverage>)

時空間オブジェクトで構成される定義域。

**備考** 連続被覆の時空間定義域は、幾何オブジェクトの集まりとの関連で定義される直接位置の集合で構成される。

#### 6.14 モザイク (tessellation)

同一次元上の隣接する時空間オブジェクトに分割したもの。

**備考** 合同で規則的な多角形や多面体で構成されるモザイクは、規則的なモザイクである。規則的だが、合同でない多角形や多面体で構成されるモザイクは、準規則的なモザイクである。これ以外のモザイクは、不規則なモザイクである。

## 6.15 不規則三角網 (TIN) (triangulated irregular network (TIN))

三角形で構成されるモザイク。

## 7. 地理識別子による空間参照のための定義

### 7.1 地名辞典 (gazetteer)

位置に関する何らかの情報を含む地物のクラスのインスタンスの要覧。

**備考** 位置情報は必ずしも座標値である必要はなく、空間参照系の記述であってもよい。

### 7.2 地理識別子 (geographic identifier)

場所の識別のための、ラベル又はコードの形式による空間参照。

**例** “日本” は、国名の例である。“102-0083” は、郵便番号の例である。

### 7.3 場所 (location)

識別可能な地理的場所。

**例** “東京タワー”，“京都市”，“北海道”。

### 7.4 空間参照 (spatial reference)

実世界における位置の記述。

**備考** これは、ラベル、コード又は座標の集合の形態を取ってもよい。

### 7.5 空間参照系 (spatial reference system)

実世界における位置を識別するための体系。

## 8. 地物カタログ化法のための定義

### 8.1 地物カタログ (feature catalogue)

一つ以上の地理データ集合に出現する地物型、地物属性及び地物関連の定義及び記述を含むカタログ。

## 9. 符号化のための定義

### 9.1 文字 (character)

データを表現、構成、又は制御するために用いられる要素の集合の構成単位。

### 9.2 コード (code)

規定された方式に従ったラベルの表現。

### 9.3 変換規則 (conversion rule)

入力データ構造をもつインスタンスを、出力データ構造をもつインスタンスに変換する方法についての仕様。

### 9.4 データ変換 (data interchange)

データの配信、受信及び解釈。



### 9.5 データ転送 (data transfer)

媒体を介し、データを一つの地点から別の地点に動かすこと。

**備考** 情報の転送はデータの転送を含む

### 9.6 符号化 (encoding)

データを一連の符号列へ変換すること。

### 9.7 符号化規則 (encoding rule)

他と識別可能な、特定のデータ構造に対する符号化を定義する変換規則の集まり。

**例** XML, ISO 10303 -21, ISO 8211

**備考** 符号化規則は、符号化すべきデータ型、符号化の結果として得られるデータ構造の中で用いられる構文、構造、及び符号化体系を規定するものである。

### 9.8 符号化サービス (encoding service)

符号化規則を実装したソフトウェアコンポーネント。

### 9.9 ファイル (file)

一つの単位として記憶又は処理される、レコードからなる名前の付いた集合。

### 9.10 インスタンスモデル (instance model)

応用スキーマに従ったデータを表現するための、概念モデル。

### 9.11 相互運用性 (interoperability)

それぞれの機能単位に固有な特性に関する知識を利用者がほとんど又は全く必要とせずに、各機能単位が互いに通信し、プログラムを実行し又はデータを転送する能力。

### 9.12 媒体 (medium)

データ構造を記憶又は伝達するための、物質又は仲介的手段。

**例** コンパクトディスク、インターネット、無線波など。

### 9.13 転送プロトコル (transfer protocol)

分散システム間の相互作用を定義するための規則の共通な集合。

## 10. UML Semantics

### 10.1 クラス (class)

同じ属性、演算、メソッド、関係及び意味を共有するオブジェクトの集合の記述。

### 10.2 オブジェクト (object)

状態と振る舞いをカプセル化した、明確に定義された境界及び識別をもつ実体。

**備考** この用語のこのような用法は、最初にオブジェクト指向プログラミングの一般理論の中で使用され、後に UML の中で同じ意味で使用されるよう採用された。オブジェクトはクラスのインスタンスである。属性と関係が状態を表す。演算、メソッド及び状態マシンは振る舞いを表す。

## 11. 略語

この規格で用いる主な略語を、次に示す。

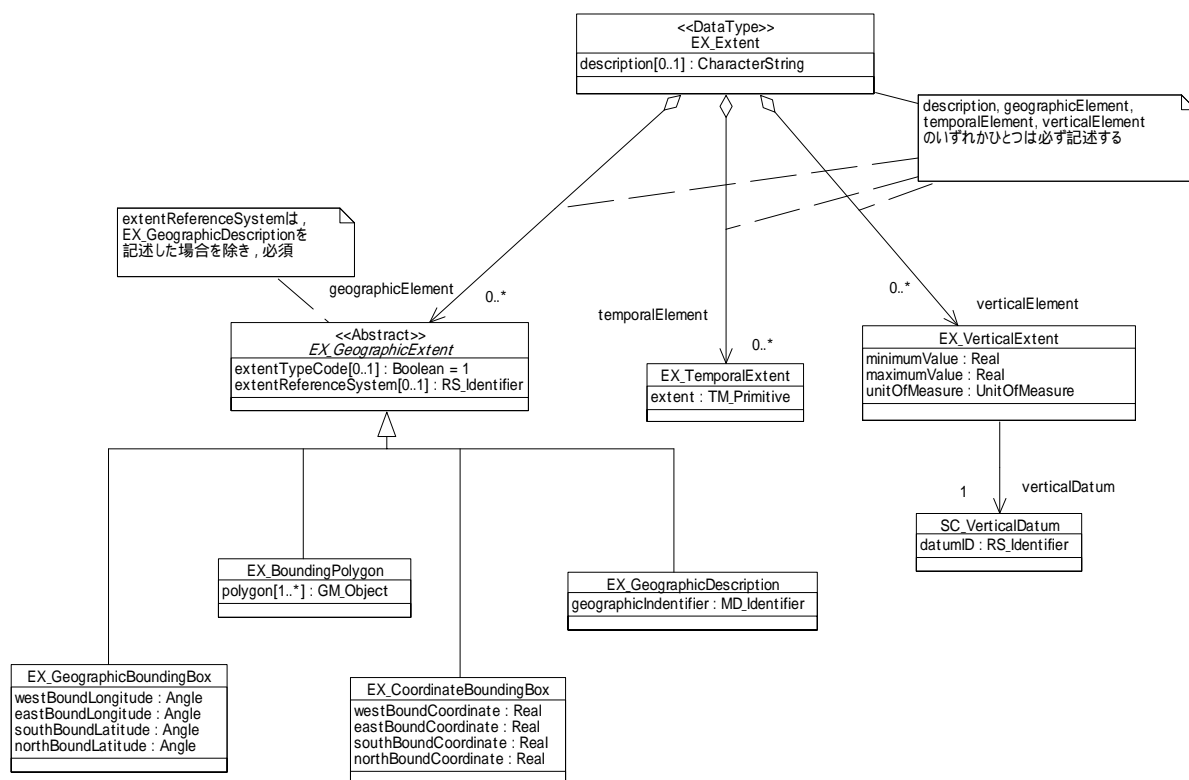
AD	西暦紀元 (Anno Domini)
BC	西暦紀元前 (Before Christ)
GIS	地理情報システム (Geographic Information System)
GPS	全地球測位システム (Global Positioning System)
TIN	不規則三角網 (Triangulated Irregular Network)
TOW	週単位の時刻 (Time of Week)
UML	UML (Unified Modeling Language )
UTC	協定世界時 (Coordinated Universal Time)
WN	週数 (Week Number)

## 附属書 6 (規定) 他の規格から引用するクラスの定義

**序文** この附属書は、JPGIS において取り上げない規格に含まれる要素で、かつ JPGIS の規定のために使用している要素を規定する。

### 1. EX\_Extent (範囲情報)

空間 (水平及び垂直) 及び時間範囲の記述のために使用する EX\_Extent のクラス図及び定義を、JMP2.0 より引用し、以下のとおり規定する。



図附 6-1 EX\_Extent

#### 1.1 EX\_Extent

範囲要素体。水平、垂直及び時間範囲についての情報を示す。

##### 属性

1. description[0..1]:CharacterString は、参照する対象の範囲。

##### 関連

1. geographicElement 関連役割は、参照している対象の水平な部分の範囲を示す。
2. temporalElement 関連役割は、参照している対象の時間的な部分の範囲を示す。
3. verticalElement 関連役割は、参照している対象の垂直的な部分の範囲を示す。

#### 1.2 EX\_GeographicExtent

水平範囲要素体。データ集合の水平範囲を示す。

## 属性

1. extentTypeCode[0..1]:Boolean は、境界多角形がデータによってカバーされる領域を示すか、データがない領域を示すかを識別する。値が“0”の場合は、領域の外部、値が“1”の場合は、領域の内部を示す。
2. extentReferenceSystem[0..1]:RS\_Identifier は、水平範囲が参照する座標参照系を示す。

### 1.3 EX\_GeographicBoundingBox

水平境界ボックス要素体。経緯度におけるデータ集合の水平境界ボックスを示す。

## 属性

1. westBoundLongitude:Angle は、東をプラスとした度による経度であらわされているデータ集合の最西端座標。
2. eastBoundLongitude:Angle は、東をプラスとした度による経度であらわされているデータ集合の最東端座標。
3. southBoundLatitude:Angle は、北をプラスとした度による緯度であらわされているデータ集合の最南端座標。
4. northBoundLatitude:Angle は、北をプラスとした度による緯度であらわされているデータ集合の最北端座標。

### 1.4 EX\_BoundingPolygon

境界ポリゴン要素体。多角形を示す座標の閉集合として表現される、データ集合を囲む境界（最終の点は最初の点と一致）を示す。

## 属性

1. polygon:GM\_Object は、境界多角形を定義する点の集合。

### 1.5 EX\_CoordinateBoundingBox

座標境界ボックス要素体。平面直角座標系におけるデータ集合の水平境界ボックスを示す。

## 属性

1. westBoundCoordinate:Real は、東をプラスとした平面直角座標であらわされているデータ集合の最西端座標。
2. eastBoundCoordinate:Real は、東をプラスとした平面直角座標であらわされているデータ集合の最東端座標。
3. southBoundCoordinate:Real は、北をプラスとした平面直角座標であらわされているデータ集合の最南端座標。
4. northBoundCoordinate:Real は、北をプラスとした平面直角座標であらわされているデータ集合の最北端座標。

### 1.6 EX\_GeographicDescription

地理的記述要素体。識別子を用いた地理的な場所の記述を示す。

## 属性

1. geographicIdentifier:MD\_Identifier は、地理的な場所を表現するための識別子。

### 1.7 EX\_TemporalExtent

時間範囲要素体。データ集合の期間を示す。

## 属性

1. extent:TM\_Primitive は、データ集合の内容への日付と時間。

## 1.8 EX\_VerticalExtent

垂直範囲要素体。データ集合の垂直範囲を示す。

### 属性

1. minimumValue:Real は、データ集合に含まれる垂直的範囲の最低値。
2. maximumValue:Real は、データ集合に含まれる垂直的範囲の最高値。
3. unitOfMeasure:UnitOfMeasure は、垂直範囲情報に使われる垂直方向の計測単位。ただしここでは、UnitOfMeasure のうち”長さ”が該当する。

### 関連

1. verticalDatum 関連役割は、最高及び最低の標高値が計測された時の原点に関する情報。

## 1.9 SC\_VerticalDatum

標高基準面原子情報。計測単位を表現するために使用する。

### 属性

1. datumID:RS\_Identifier は、原子の識別子。RS\_Identifier は、附属書 2 2.2 を参照。

## 2. ReferenceSystemInformation (参照系情報)

参照系の情報のうち、範囲情報から参照される MD\_Identifier のクラス図及び定義を、JMP2.0 より引用し、以下のとおり規定する。

MD_Identifier
authority[0..1]:CI_Citation code:CharacterString

図附 6-2 MD\_Identifier

## 2.1 MD\_Identifier

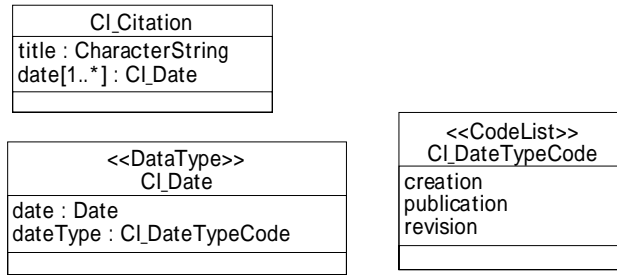
メタデータ識別子要素体。名前空間において一意のコード値を与えるクラスを示す。

### 属性

1. authority[0..1]:CI\_Citation は、名前空間の保守責任を有する個人又は団体。
2. code:CharacterString は、名前空間のインスタンスを識別する文字の値。

## 3. CI\_Citation (引用情報)

情報資源への参照を記述のために使用する CI\_Citation のクラス図及び定義を、JMP2.0 より引用し、以下のとおり規定する。



図附 6-3 CI\_Citation

### 3.1 CI\_Citation

引用要素体。情報資源への参照を示す。

#### 属性

1. title:CharacterString は、引用された情報資源を示す名称。
2. date[1..\*]:CI\_Date は、引用された情報資源を参照した日付。

### 3.2 CI\_Date

日付要素体。参照用の日付及びその記述に使用される事象。

#### 属性

1. date:Date は、引用された情報資源の参照用の日付。
2. dateType:CI\_DateTypeCode は、日付の参照に使用される事象。

### 3.3 CI\_DateTypeCode

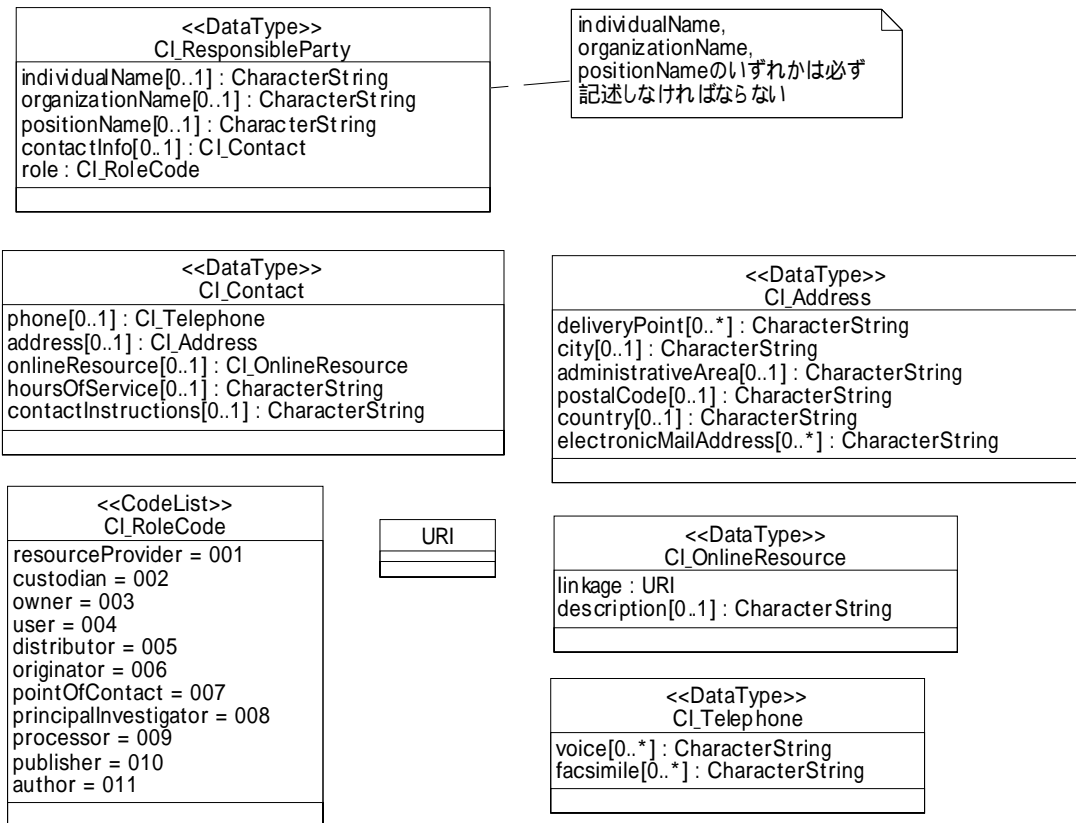
日付型コード。イベントが発生したときに関わる識別情報。符号リストで、以下の値をもつ。

#### 符号リスト

- a) creation=001 : 作成日。情報資源が作成された日付を示す識別情報。
- b) publicaation=002:刊行日。情報資源が発表された日付を示す識別情報。
- c) revision=003:改訂日。情報資源が、調査又は再調査及び改訂又は補足された日付を示す識別情報。

## 4. CI\_ResponsibleParty (責任者情報)

責任者情報 CI\_ResponsibleParty のクラス図及び定義を、JMP2.0 より引用し、以下のとおり規定する。



図附 6-4 CI\_ResponsibleParty

#### 4.1 CI\_ResponsibleParty

責任者要素体。データ集合に関連した人と組織の識別情報及び連絡手段を示す。

##### 属性

1. individualName[0..1]:CharacterString は、責任をもつ個人の苗字、名前、所属。
2. organizationName[0..1]:CharacterString は、責任をもつ組織の名称。
3. positionName[0..1]:CharacterString は、責任をもつ個人の役割又は役職の名称。
4. contactInfo[0..1]:CI\_Contact は、責任者の情報。
5. role:CI\_RoleCode は、責任者の果たす機能。

#### 4.2 CI\_Contact

問合せ先要素体。責任をもつ個人又は組織に問合せするための情報を示す。

##### 属性

1. phone[0..1]:CI\_Telephone は、問合せされる組織又は個人の電話番号。
2. address[0..1]:CI\_Address は、問合せされる組織又は個人の住所及び電子メールアドレス。
3. onlineResource[0..1]:CI\_OnlineResource は、個人又は組織への問合せに使われるオンライン情報。
4. hoursOfService[0..1]:CharacterString は、個人が組織又は個人に問合せることができる時間。
5. contactInstructions[0..1]:CharacterString は、個人又は組織へ問合せのための補足

的な手引き。

### 4.3 CI\_Address

住所要素体。責任をもつ個人又は組織の住所を示す。

#### 属性

1. deliveryPoint[0..\*]:CharacterString は、場所を示す住所。
2. city[0..1]:CharacterString は、場所の市町村名称。
3. administrativeArea[0..1]:CharacterString は、都道府県の名称。
4. postalCode[0..1]:CharacterString は、郵便番号。
5. country[0..1]:CharacterString は、国名。
6. electronicMailAddress[0..\*]:CharacterString は、責任をもつ組織又は個人の電子メールアドレス。

### 4.4 CI\_OnlineResource

オンライン情報資源要素体。データ集合、仕様あるいは共用のプロファイル名称及び拡張メタデータ要素が得られるオンライン情報資源に関する情報を示す。

#### 属性

1. linkage:URI は、“http://www.statkart.no/isotc211”のようなユニフォーム・リソース・アイデンティファイアー (URI) 又は類似のアドレス指定法によるオンラインアクセス用の所在 (住所)。
2. description[0..1]:CharacterString は、オンライン情報資源が何か、又は何をやるかの記述。

### 4.5 URI

ユニフォーム・リソース・アイデンティファイアー (Uniform Resource Identifiers) を示す。

### 4.6 CI\_Telephone

電話要素体。責任をもつ個人又は組織に問合せをするための電話番号とファクシミリ番号。

#### 属性

1. voice[0..\*]:CharacterString は、責任をもつ個人又は組織の電話番号。
2. facsimile[0..\*]:CharacterString は、責任をもつ個人又は組織のファクシミリ番号。

### 4.7 CI\_RoleCode

役割コード。責任者によって実行される機能。

#### 符号リスト

- a) resourceProvider: 情報資源提供者。情報資源を供給する団体。
- b) custodian: 管理者。情報資源の適切な管理、保守を保証し、データのために説明責任、管理責任を負う団体。
- c) owner: 所有者。情報資源を所有している団体。
- d) user: 利用者。情報資源を使用する団体。
- e) distributor: 配布者。情報資源を配布する団体。
- f) originator: 創作者。情報資源を作成した団体。
- g) pointOfContact: 問合せ先。情報資源を得るため、又はその知識を得るための問合せ可能とする団体。
- h) principalInvestigator: 主要な調査担当者。情報を集め、研究の指揮をとる主要な団

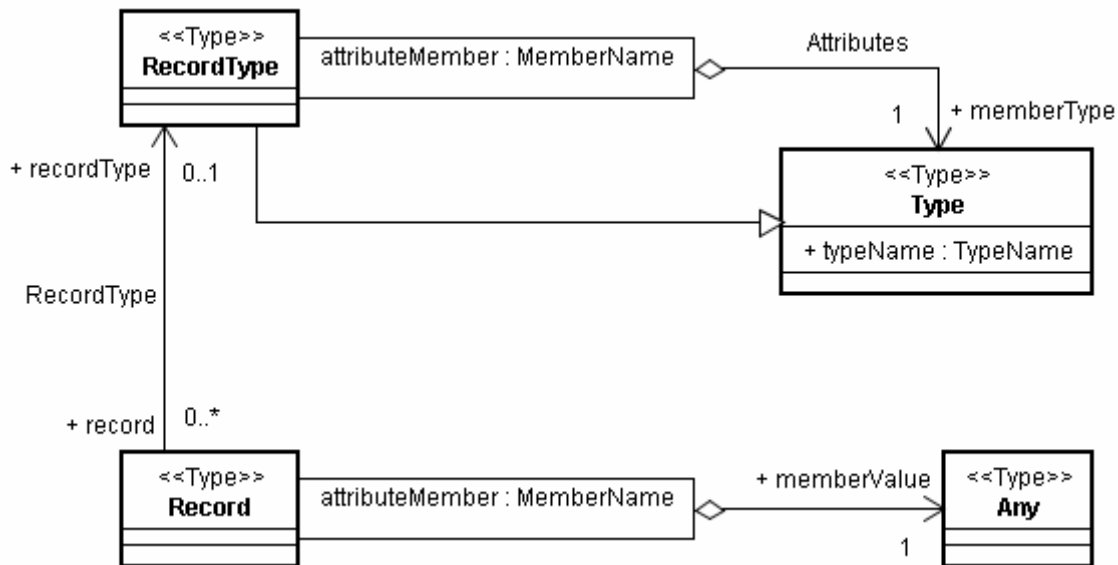


体。

- i) processor : 処理担当者。情報資源を修正して、データを処理した団体。
- j) publisher : 刊行者。情報資源を刊行した団体。
- k) author : 著作者。情報資源を著作した団体。

## 5. Record 及び RecordType

値を記述するために使用される Record 及び RecordType のクラス図及び定義を、ISO/TS 19103-Conceptual Schema Language より引用する。



図附 6-5 Record 及び RecordType

### 5.1 RecordType

RecordType は、Record が表す属性値の型を示す。

#### 属性

1. [上位から継承する属性] typeName:CharacterString は、RecordType 型の名称。

#### 関連

1. memberType 関連役割は、属性名 attributeMember:MemberName によって限定され、その属性名に対応する型を示す。
2. record 関連役割は、この RecordType を参照する Record 型を示す。

### 5.2 Record

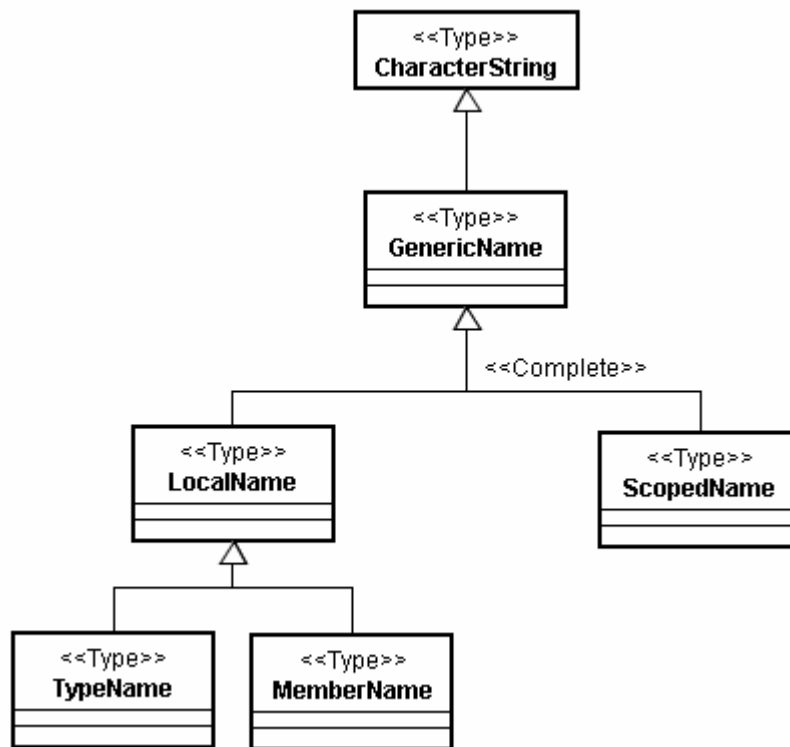
Record は、属性値を表すための型であり、属性名と実際の値を対にして表すことを基本とする。

#### 関連

1. recordType 関連役割は、Record が参照する型を示す。
2. memberValue 関連役割は、属性名 attributeMember:MemberName によって限定され、その属性の実際の値を示す。

## 6. Names (名前情報)

クラス名、属性名など、名前を記述するためにこのプロファイルで使用する型の定義を ISO/TS 19103-Conceptual Schema Language より引用する。なお、ここでは ISO 19103 には明記されていない `CharacterString` 型から `GenericName` 型への継承を記述しているが、これは、JPGIS では操作に関する定義をすべて割愛しているため、本来の定義の意味を維持するための付加である。



図附 6-6 名前型

### 6.1 GenericName

一般名称を表す型。

URL 名や名前空間名などの場合、区切り文字を用いて複数の名前が連結される。その場合、局所部分での識別子となる名前を `LocalName`、その他の部分を `ScopedName` と分類する。

### 6.2 ScopedName

`GenericName` のなかで、`LocalName` を切り出した残りの部分を表す型。

### 6.3 LocalName

限定した範囲で有効なオブジェクトの名称を表す型。

### 6.4 TypeName

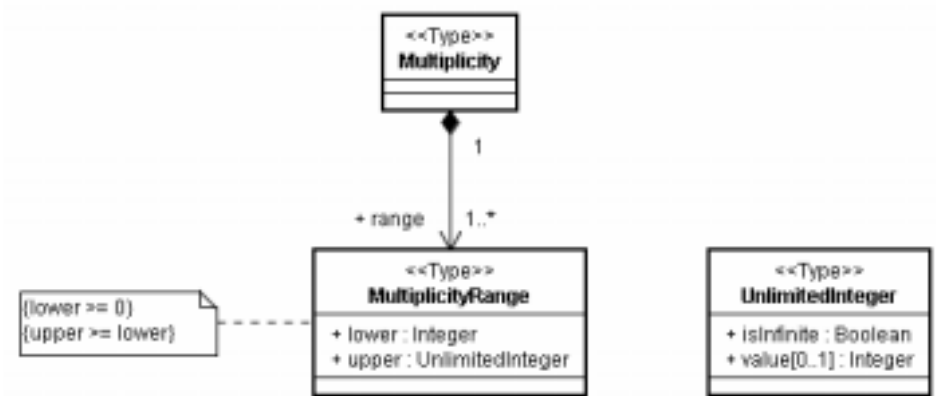
`LocalName` のなかで、レコード型又はオブジェクト型の名称を表す型。

### 6.5 MemberName

`LocalName` のなかで、属性、関連等の名称を表す型。

## 7. Multiplicity (多重度情報)

一般地物モデルを定義する際、多重度を記述するために参照する型の定義を ISO/TS 19103-Conceptual Schema Language より引用する。



図附 6-7 Multiplicity

### 7.1 Multiplicity

クラス図において、属性や関連役割の多重度を表す型。

#### 関連

1. range 関連は、多重度を数値の範囲で示すための合成である。

### 7.2 MultiplicityRange

多重度の範囲を定義する型。

#### 属性

1. lower:Integer は、多重度の範囲の下限値を整数値で示す。
2. upper:UnlimitedInteger は、多重度の範囲の上限値を示す。

#### 制約

1. lower の値は、0 以上の整数でなければならない。
2. upper の値は、lower の値以上（無制限を含む）でなければならない。

### 7.3 UnlimitedInteger

多重度の上限値を定義する型。

#### 属性

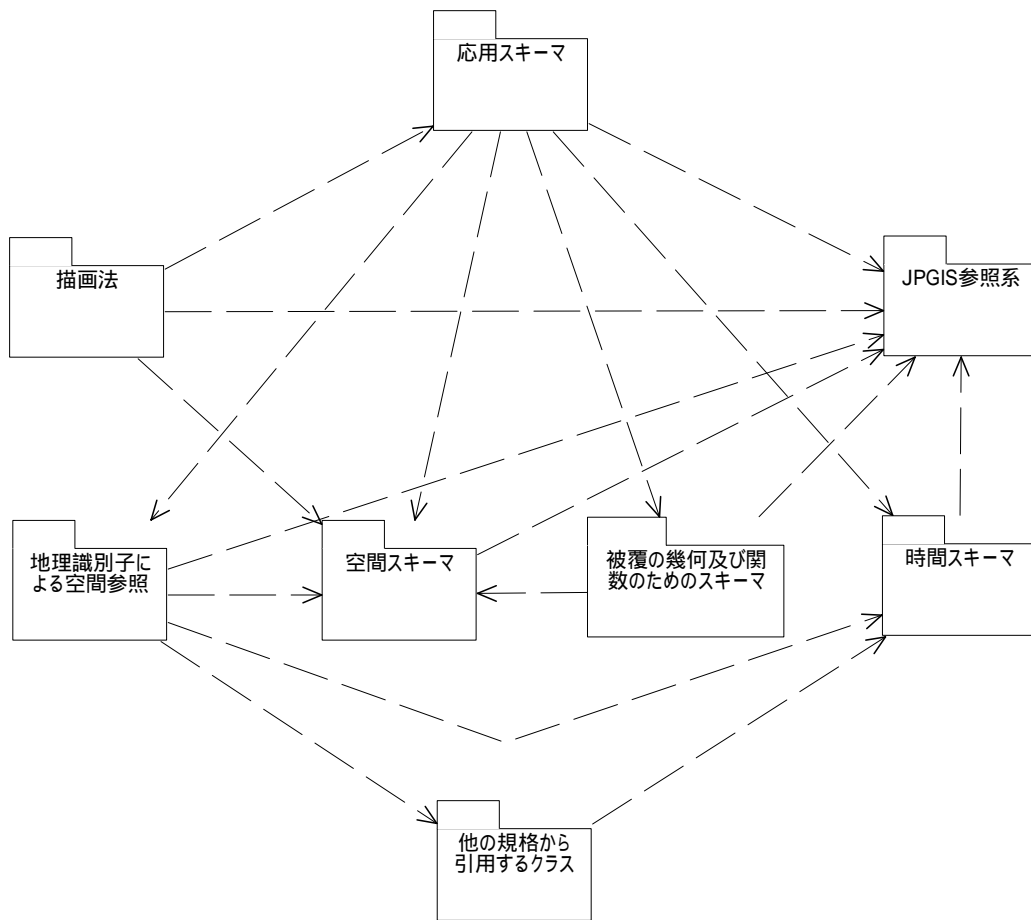
1. isInfinite:Boolean は、多重度の範囲の上限値が無制限か否かを示す。
2. value[0..1]:Integer は、多重度の範囲の最大値を整数値で示す。

## 附属書 7 (参考) 参照モデル

**序文** この附属書は、このプロファイルの全体的な枠組みを示す参照モデルであり、規定の一部ではない。

### 1. パッケージ依存関係

このプロファイルの全体的な枠組みは、図附 7-1に示すパッケージ図のとおりである。



図附 7-1 参照モデル

#### 1.1 応用スキーマパッケージ

応用スキーマパッケージは、このプロファイルで規定する応用スキーマのための規則に従い、応用システムのために作成された応用スキーマを格納するためのパッケージである。

応用スキーマパッケージは、このプロファイルで規定する以下のパッケージに依存する。

- 空間スキーマ
- 時間スキーマ
- 被覆の幾何及び関数のためのスキーマ
- 地理識別子による空間参照

- ・ JPGIS 参照系

応用スキーマパッケージに格納される応用スキーマは、依存するパッケージで規定する要素を利用することができる。

## 1.2 空間スキーマパッケージ

空間スキーマパッケージは、このプロファイルで規定する空間スキーマを格納するためのパッケージである。空間スキーマパッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ JPGIS 参照系

## 1.3 時間スキーマパッケージ

時間スキーマパッケージは、このプロファイルで規定する時間スキーマを格納するためのパッケージである。時間スキーマパッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ JPGIS 参照系

## 1.4 被覆の幾何及び関数のためのスキーマパッケージ

被覆の幾何及び関数のためのスキーマパッケージは、このプロファイルで規定する被覆の幾何及び関数のためのスキーマを格納するためのパッケージである。

被覆の幾何及び関数のためのスキーマパッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ 空間スキーマ
- ・ JPGIS 参照系

被覆の幾何及び関数のためのスキーマパッケージに格納される被覆の幾何及び関数のためのスキーマが、依存するパッケージで規定するどの要素を利用するかについては、このプロファイルの本文に記述する。

## 1.5 地理識別子による空間参照パッケージ

地理識別子による空間参照パッケージは、このプロファイルで規定する地理識別子による空間参照を格納するためのパッケージである。

地理識別子による空間参照パッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ 空間スキーマ
- ・ 時間スキーマ
- ・ 他の規格から引用するクラス
- ・ JPGIS 参照系

地理識別子による空間参照パッケージに格納される地理識別子による空間参照が、依存するパッケージで規定するどの要素を利用するかについては、このプロファイルの本文に記述する。

## 1.6 描画法パッケージ

描画法パッケージは、このプロファイルの附属書 10 で示す描画スキーマを格納するためのパッケージである。

描画法パッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ 空間スキーマ
- ・ 応用スキーマ

- ・ JPGIS 参照系

描画法パッケージに格納される描画スキーマが、依存するパッケージで規定するどの要素を利用するかについては、このプロファイルの附属書 10 に記述する。

### 1.7 他の規格から引用するクラスパッケージ

他の規格から引用するクラスパッケージは、このプロファイルで利用する他の規格から引用するクラスのスキーマを格納するためのパッケージである。

他の規格から引用するクラスパッケージは、このプロファイルで規定する以下のパッケージに依存する。

- ・ 時間スキーマ

他の規格から引用するクラスパッケージに格納されるスキーマが、依存するパッケージで規定するどの要素を利用するかについては、このプロファイルの附属書 6 に記述する。

### 1.8 JPGIS 参照系パッケージ

JPGIS 参照系パッケージは、このプロファイルの附属書 2 で規定する時間及び空間参照系スキーマを格納するためのパッケージである。

## 2. dependency matrix

このプロファイルのパッケージ依存関係を、以下のとおり dependency matrix で示す。

表附 7-1 dependency matrix

パッケージ名	応用スキーマ	空間スキーマ	時間スキーマ	被覆の幾何及び関数のためのスキーマ	地理識別子による空間参照	描画法	他の規格から引用するクラス	JPGIS 参照系
応用スキーマ		X	X	X	X			X
空間スキーマ								X
時間スキーマ								X
被覆の幾何及び関数のためのスキーマ		X						X
地理識別子による空間参照		X	X				X	X
描画法	X	X						X
他の規格から引用するクラス			X					
JPGIS 参照系								

※ "X"が依存関係の存在を示す。

横軸の項目から縦軸の項目に向かって、依存関係の有無を表現している。

## 附属書 8 (参考) XML に基づく符号化規則

**序文** この附属書では、システムやプラットフォームに依存しないデータ交換に用いる、XML に基づく符号化規則を明示する。この附属書は、「12 符号化」に従い、以下の項目を明示する。この附属書は、このプロファイルの「12 符号化」に基づいて作成した XML に基づく符号化規則を示すものであり、規定の一部ではない。

1. 符号化要件 (附属書 8 1)
2. 入力データ構造 (附属書 8 2)
3. 出力データ構造 (附属書 8 3)
4. 変換規則 (附属書 8 4)
5. JPGIS 標準スキーマの XML Schema (附属書 8 5)

### 1. 符号化要件

#### 1.1 応用スキーマとスキーマ言語

このプロファイルに準拠する応用スキーマは、「6 応用スキーマのための規則」における規則に基づき、OMG UML1.4.2 で記述する。

#### 1.2 文字レパートリと言語

この附属書に基づく符号化規則では、「12.3.2.2 文字レパートリと言語」に従って文字レパートリを使用する。ただし、附属書 8 5 に規定する JPGIS 標準スキーマの W3C XML Schema による記述は、UTF-8 を用いる。

**例** 文字レパートリの記述例は次のとおりである。この情報は、W3C XML Schema 及びそれに従って作成されるインスタンス文書のヘッダに記述しなければならない。

```
<?xml version="1.0" encoding="UTF-8"?>
```

#### 1.3 データ集合とオブジェクト識別

この附属書に基づく符号化規則では、以下の二つのオブジェクト識別を規定する。

1. XML の ID メカニズムに基づくデータ集合で一意となるオブジェクト識別子
2. アプリケーション定義域仕様に基づく普遍的で一意的な識別子 (UUID)

### 2. 入力データ構造

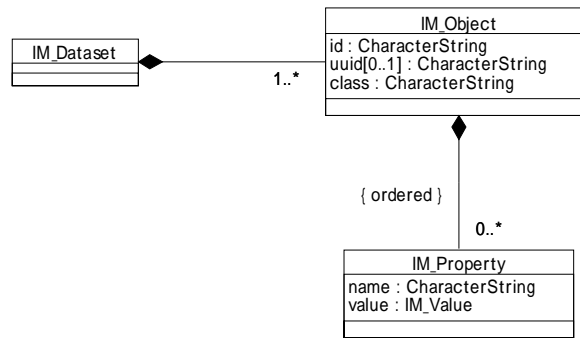
#### 2.1 一般的なインスタンスモデル

インスタンスモデルは、UML で表現された応用スキーマに従い記述されたデータを表すことができる。インスタンスモデルは、応用スキーマ特有のデータ構造に従う。インスタンスモデルは、一連のオブジェクト (IM\_Object) を含むデータ集合 (IM\_Dataset) から成り、オブジェクトは一連のプロパティを含む。それぞれのプロパティはその型に従って符号化される。インスタンスモデルを図附 8-1 に示す。

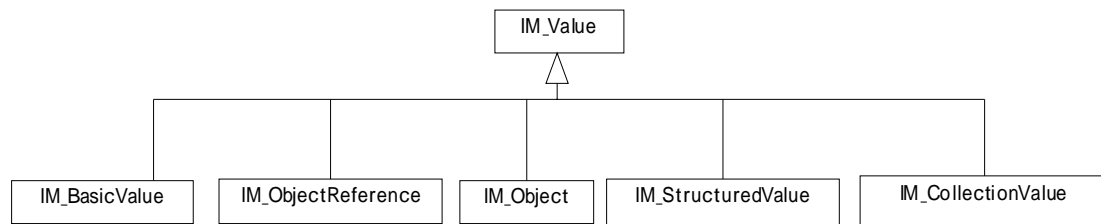
データ集合に含まれる情報の基本単位はオブジェクトである。オブジェクトは単一の具体的な型でなければならない。従って、《Abstract》や《Interface》をステレオタイプとしてもつ型は符号化してはならない。

オブジェクトは、応用スキーマで定義された、それと一致する型をもち、とりうる属性と関連を定義する。IM\_Object は“class”属性によりその型を参照し、データ集合の中で、一意となる識別子“id”により識別しなければならない。また IM\_Object は“uuid”属性によりアプリケーション・ドメインや名前空間全体の中で一意に識別することができる。

型に定義された属性と関連は順序づけられたプロパティ列に展開される。プロパティ (IM\_Property) は名前と値の組み合わせを与え、属性、関連、集成又は合成で表すことができる。“name”属性により、その属性や関連を参照するため、プロパティ名は属性名又は関連の役割名と一致しなければならない。値 (IM\_Value) はプロパティの値を表す。



図附 8-1 一般的なインスタンスモデル



図附 8-2 インスタンスモデル：値型

値 (IM\_Value) は、以下のように五つの型で定義される (図附 8-2)。

- IM\_BasicValue は基本型の値として表される。例えば Integer や CharacterString である。
- IM\_ObjectReference は目標オブジェクトへのリンク又は参照を表す。目標オブジェクトは同じデータ集合、又は他のデータ集合の中にあってもよい。一意の識別子 (ID) は、同じデータ集合に含まれるオブジェクトを対象とする。UUID はアプリケーション領域の中にあるオブジェクトを対象とする。
- IM\_Object はオブジェクトを表す。
- IM\_StructuredValue はプロパティ列 (IM\_Property) のような属性の集合を表す。
- IM\_CollectionValue は、Set, Sequence や Dictionary の様な、コレクションデータ型を表す。

インスタンスモデルと OMG UML1.4.2 のステレオタイプの対応付けを附属書 8.2.2 に示す。

## 2.2 UML とインスタンスモデルの関係

表附 8-1は UML のクラスとインスタンスモデルのクラスの関係を表したものである。N/A は適用外である。



表附 8-1 UML とインスタンスモデルの関係

UML 概念	インスタンスモデル
パッケージ	N/A
Application Schema	N/A
Leaf	N/A
Package	N/A
クラス	
ステレオタイプ	
<<Abstract>>	N/A
<<Codelist>>	IM_BasicValue
<<DataType>>	IM_StructuredValue
<<Enumeration>>	IM_BasicValue
<<Feature>>	IM_Object
<<Interface>>	N/A
<<Metaclass>>	N/A
<<Record>>	IM_CollectionValue
<<Type>>	IM_Object
<<Union>>	IM_StructuredValue
属性	データ型 (IM_BasicValue, IM_Object 又は IM_StructuredValue) に従った IM_Value をもつ IM_Property
関連	IM_ObjectReference 又は IM_Object としての IM_Value をもつ IM_Property

### 3. 出力データ構造

#### 3.1 XML 文書

この附属書で示す符号化規則は XML1.0 に準拠している。XML はテキストフォーマットで記述され、すべてのデータ型は文字列へ符号化しなければならない。データは、XML 勧告に従って符号化しなければならない。

交換フォーマットの構造を決定付ける出力データ構造のためのスキーマは、有効な W3C XML Schema による記述でなければならない。

#### 3.2 XML Schema 文書 (XSD)

この附属書で示す符号化規則では、XML 文書のデータ・インスタンスを明示する要素宣言を W3C XML Schema による記述で定義する。XML Schema は附属書 8 4 に示す変換規則に従って定義しなければならない。

### 4. 変換規則

#### 4.1 序文

ここでは、UML で表現された応用スキーマから、W3C XML Schema 文書をどのように生成するかを定義するスキーマ変換規則を示す。また、W3C XML Schema 文書からどのように XML 文書を生成するかを定義するインスタンス変換規則についても、一部記述する。

#### 4.2 名前空間

この附属書に則って作成される W3C XML Schema は、W3C XML Schema の要素や属性が属する名前空間及び要素や属性に付するための名前空間を規定しなければならない。

### 4.3 オブジェクト識別子

応用スキーマに定義した型のうち、IM\_Object に対応する型は、オブジェクト識別子をもつ構造をとらなくてはならない。オブジェクト識別子は以下のように定義する。

```
<xs:attributeGroup name="IM_ObjectIdentification">
  <xs:attribute name="id" type="xs:ID"/>
  <xs:attribute name="uuid" type="xs:string"/>
</xs:attributeGroup>

<complexType name="IM_Object">
  <attributeGroup ref="IM_ObjectIdentification"/>
</complexType>
```

IM\_Object はオブジェクトを識別するために使う二つの属性をもつ。“id”属性はひとつの XML 文書の中で固有でなければならず、XML の“ID”データ型の定義に従わなければならない。

“uuid”属性はひとつのアプリケーションの範囲の中で普遍的に定義され、普遍的に固有な識別子を提供するために用いてもよい。

### 4.4 オブジェクト参照

IM\_ObjectReference は、オブジェクトを参照するために用いる型であり、四つの属性をもつ。“idref”属性は、同一のデータ集合の XML 要素によって与えられるオブジェクトを参照するために用いなければならない。“uuidref”属性はアプリケーション領域において、オブジェクトを参照するために用いなければならない。“uriref”属性は、他のデータ集合に存在するオブジェクトを参照するために用いなければならない。その表現は URI を用いる。”xlink:simpleLink”属性グループは、xlink を使用してオブジェクトを参照するために用いなければならない。IM\_ObjectReference の型のインスタンスは、与えられる四つの属性のうち、一つだけもつことができる。

```
<xs:attributeGroup name="IM_ObjectReference">
  <xs:attribute name="idref" type="xs:IDREF"/>
  <xs:attribute name="uuidref" type="xs:string"/>
  <xs:attribute name="uriref" type="URI"/>
  <xs:attributeGroup ref="xlink:simpleLink"/>
</xs:attributeGroup>

<xs:complexType name="IM_ObjectReference">
  <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
```

### 4.5 基本的なデータ型

#### 4.5.1 基本データ型

基本データ型は、W3C XML Schema の単純型宣言に変換しなければならない。W3C XML Schema で定義されているどのデータ型も使用することができる。基本データ型の符号化は、W3C XML Schema Part 2:Datatypes の中で定義された正規化表現に従わなければならない。

このプロファイルで定義された基本データ型 (JPGIS 5.1.1) は、以下の通り変換される。

**備考** ユーザは、W3C XML Schema に定義された制限の仕組みを使用して、基本データ型に制限を加えることができる。

#### 4.5.1.1 Number

Number 型は抽象型であり、整数型、実数型及び十進数型の上位型として、このプロファイルでは定義している。しかし、W3C XML Schema は抽象的な Number を定義せず、decimal のデータ型に基づく単純型であることを宣言する。

```
<xs:simpleType name="Number">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

#### 4.5.1.2 Integer

Integer 型は W3C XML Schema の int データ型に基づく。

```
<xs:simpleType name="Integer">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
```

#### 4.5.1.3 Decimal と Real

Decimal 型及び Real 型は、W3C XML Schema の decimal 型に基づく。

```
<xs:simpleType name="Decimal">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

```
<xs:simpleType name="Real">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

#### 4.5.1.4 Vector

Vector 型は数字の順序付けられた集合として定義する。decimal のリストで構成する。

```
<xs:simpleType name="Vector">
  <xs:list itemType="xs:decimal"/>
</xs:simpleType>
```

#### 4.5.1.5 CharacterString

CharacterString 型は、W3C XML Schema の string 型に基づき、ISO 10646-1 のどのような文字も表すことができる。また、12.3.2.2 に規定した文字レパートリを使用することができる。

```
<xs:simpleType name="CharacterString">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

#### 4.5.1.6 Date

Date 型は、W3C XML Schema の date 型に基づき、JIS X 0301 に準拠して符号化する。

```
<xs:simpleType name="Date">
  <xs:restriction base="xs:date"/>
</xs:simpleType>
```

#### 4.5.1.7 Time

Time 型は、W3C XML Schema の time 型に基づき、JIS X 0301 に準拠して符号化する。

```
<xs:simpleType name="Time">
  <xs:restriction base="xs:time"/>
</xs:simpleType>
```

#### 4.5.1.8 DateTime

DateTime 型は、W3C XML Schema の dateTime 型に基づき、JIS X 0301 に準拠して符号化する。

```
<xs:simpleType name="DateTime">
  <xs:restriction base="xs:dateTime"/>
</xs:simpleType>
```

#### 4.5.1.9 Boolean

Boolean 型は、W3C XML Schema の boolean 型に基づく。値は論理的な偽である” 0” 又は” false” をとるか、あるいは論理的な真である” 1” 又は” true” をとる。

```
<xs:simpleType name="Boolean">
  <xs:restriction base="xs:boolean"/>
</xs:simpleType>
```

### 4.5.2 コレクションデータ型

コレクションデータ型はパラメタが、実引数に結び付けられる型である。JPGIS の 5.1.2 では三つの異なるコレクションデータ型を定義している（セット型（Set<T>）、列型（Sequence<T>）、及び辞書型（Dictionary<K,V>）である。最初の二つは、一つのパラメタを利用する。辞書型は二つのパラメタを利用する。

- コレクションデータ型は、テンプレート型に相当する複合型定義に変換しなければならない。型宣言が指定されなければならない。その名称は、下線 “\_” 文字で分離された引数名とテンプレート名を連結することにより構築してもよい。
- セット型、列型は、複合型定義に写像しなければならない。それは、単一の引数の型に従って名前付けや型付けされた一つの要素を含む無限の多重度で構成する列で定義される。引数の型が基本型である場合、XML Schema のリスト構成を使用した単純型定義とする。
- 辞書型は、複合型定義に写像しなければならない。それは、二つの引数の型に従って名前付

けや型付けされた二つの要素を含む無限の多重度で構成する列で定義される。

- セット型、列型のパラメタが基本データ型以外の場合

```
<xs:complexType name="Sequence_TypeName_">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="TypeName"/>
  </xs:sequence>
</xs:complexType>
```

- セット型、列型のパラメタが基本データ型の場合

```
<xs:simpleType name="Sequence_BasicTypeName_">
  <xs:list itemType="BasicTypeName"/>
</xs:simpleType>
```

- 辞書型

```
<xs:complexType name="Dictionary_AttributeName_Any_">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="AttributeName" type="CharacterString"/>
    <xs:element name="Any" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="Dictionary_AttributeName_TypeName_">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="AttributeName" type="CharacterString"/>
    <xs:element name="TypeName" type="CharacterString"/>
  </xs:sequence>
</xs:complexType>
```

### 4.5.3 測定量及び測定単位

#### 4.5.3.1 Measure

Measure は、W3C XML Schema の複合型定義として変換しなければならない。

```
<xs:complexType name="Measure">
  <xs:sequence>
    <xs:element name="value" type="Number"/>
    <xs:element name="uom" type="UnitOfMeasure" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

以下の単位を用いる場合は、単位の記述を省略してもよい。

- 長さ：メートル
- 時間：秒
- 角度：ラジアン
- 面積：平方メートル
- 体積：立方メートル
- 速度：メートル毎秒

#### 4.5.3.2 UnitOfMeasure

UnitOfMeasure は、W3C XML Schema の複合型定義として変換しなければならない。

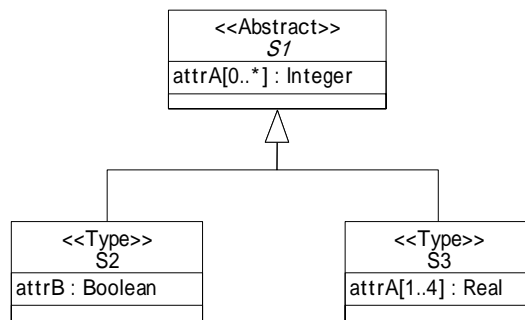
```
<xs:complexType name="UnitOfMeasure">
  <xs:sequence>
    <xs:element name="uomName" type="CharacterString"/>
  </xs:sequence>
</xs:complexType>
```

### 4.6 ステレオタイプ

応用スキーマに定義した型は、そのステレオタイプに応じて以下のように W3C XML Schema に変換しなくてはならない。

#### 4.6.1 抽象型 (Abstract)

ステレオタイプが《Abstract》となる型は”abstract”属性を”true”とした複合型定義に変換しなければならない。



図附 8-3 抽象型の記載例

```
<xs:complexType name="S1" abstract="true">
  <xs:complexContent>
    <xs:extension base="IM_Object">
      <xs:sequence>
        <xs:element name="attrA" type="Integer" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
```

```
</xs:complexType>
```

#### 4.6.2 符号リスト型 (CodeList)

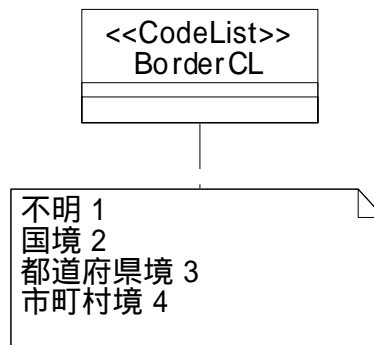
ステレオタイプが《CodeList》となる型の宣言は、複合型定義に変換しなければならない。符号リストの W3C XML Schema を以下に示す。

```
<complexType name="codelist">
  <sequence maxOccurs="unbounded">
    <element name="description" type="xs:string"/>
    <element name="codelabel" type="codevalue"/>
  </sequence>
  <attribute name="name" type="xs:string"/>
  <attribute name="version" type="xs:string" use="optional"/>
  <attribute name="date" type="xs:date" use="optional"/>
</complexType>
```

```
<complexType name="codevalue">
  <attribute name="label" type="xs:string"/>
  <attribute name="code" type="xs:string"/>
</complexType>
```

応用スキーマで符号リストを定義した場合は、この複合型を拡張して定義する。

**例** 境界符号リスト (BorderCL)



図附 8-4 コードリスト型の例

[W3C XML Schema]

```
<xs:complexType name="BorderCL" >
  <xs:complexContent>
    <xs:extension base="codelist"/>
  </xs:complexContent>
</xs:complexType>
```

[W3C XML Schema に基づくインスタンス文書]

```

<BorderCL version="2.3.4" date="2001-05-18">
  <description>異なる境界の型</description>
  <codelabel label="不明" code="1"/>
  <codelabel label="国境" code="2"/>
  <codelabel label="都道府県境" code="3"/>
  <codelabel label="市町村境" code="4"/>
</BorderCL >

```

- 符号リストの使用方法

応用スキーマにおいて符号リストを属性の型として定義する場合、以下に示す複合型を、属性の要素宣言の type に指定する。

```

<xs:complexType name="CodeType">
  <xs:simpleContent>
    <xs:extension base="string"/>
    <xs:attribute name="codespace" type="anyURI" use="optional"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>

```

#### 4.6.3 データ型 (DataType)

ステレオタイプが《DataType》となる型は、W3C XML Schema の複合型定義として変換しなければならない。属性や関連は、XML の属性宣言か、又はローカルな XML 要素で宣言しなければならない。属性や関連の要素の順序は、複合型定義の中で与える。

<<DataType>> TM_TemporalPosition
indeterminatePosition[0..1] : TM_IndeterminateValue

図附 8-5 データ型の例

```

<complexType name=" TM_TemporalPosition">
  <sequence>
    <element name="indeterminatePosition" type="TM_IndeterminateValue"
      minOccurs="0"/>
  </sequence>
</complexType>

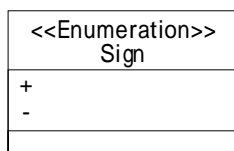
```

#### 4.6.4 列挙型 (Enumeration)

ステレオタイプが《Enumeration》となる型は、複数の列挙された値をもつ文字列に、制約をかけた単純型に変換しなければならない。

**例** 文字に基づいた列挙型の Sign クラスの場合、値は “+” 又は “-” の制約を与えている。



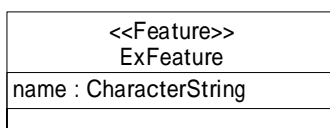


図附 8-6 列挙型の例

```
<xs:simpleType name="Sign">
  <xs:restriction base="xs:string">
    <xs:enumeration value="+"/>
    <xs:enumeration value="-"/>
  </xs:restriction>
</xs:simpleType>
```

#### 4.6.5 地物型 (Feature)

ステレオタイプが《Feature》となる型、又はステレオタイプの指定がない型は、クラスと同じ名前を備えた複合型の定義に写像しなければならない。複合型定義は、IM\_Object からの継承、又は IM\_ObjectIdentificaton 属性の集合への参照を通じた識別属性を含まなければならない。この写像の結果は、《Type》型と等しい。



図附 8-7 地物型の例

- IM\_Object からの継承による記述例

```
<xs:complexType name="ExFeature">
  <xs:complexContent>
    <xs:extension base="IM_Object">
      <xs:sequence>
        <xs:element name="name" type="CharacterString"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

- IM\_ObjectIdentification 属性の集合への参照による記述例

```
<xs:complexType name="ExFeature">
  <xs:sequence>
    <xs:element name="name" type="CharacterString"/>
  </xs:sequence>
  <attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
```

#### 4.6.6 レコード型 (Record)

ステレオタイプが「Record」となる型の宣言は、XML の複合型定義に変換しなければならない。レコード型の W3C XML Schema を以下に示す。

```
<xs:complexType name="Record">
  <xs:sequence>
    <xs:element name="attributeMember" type="MemberName"
      minOccurs="0"/>
    <xs:element name="memberValue" type="xs:anyType"/>
    <xs:element name="recordType" type="ref_RecordType"/>
  </xs:sequence>
  <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
<xs:complexType name="RecordType">
  <xs:complexContent>
    <xs:extension base="IM_Object">
      <xs:sequence>
        <xs:element name="typeName" type="TypeName"/>
        <xs:element name="attributeMember" type="MemberName"
          minOccurs="0"/>
        <xs:element name="memberType" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="typeName" type="TypeName"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="record" type="jps:ref_Record" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="IM_ObjectIdentification"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ref_RecordType">
  <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
<xs:complexType name="ref_Record">
  <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
```

#### 4.6.7 タイプ型 (Type)

ステレオタイプが《Type》となる型、又はステレオタイプの指定がない型は、クラスと同じ名前を備えた複合型の定義に写像しなければならない。複合型定義は、IM\_Object からの継承、又は IM\_ObjectIdentificaton 属性の集合への参照を通じた識別属性を含まなければならない。この写像の結果は、《Feature》型と等しい。

<<Type>> ExType
name : CharacterString

図附 8-8 Type 型の例

- IM\_Object からの継承による記述例

```
<xs:complexType name="ExType">
  <xs:complexContent>
    <xs:extension base="IM_Object">
      <xs:sequence>
        <xs:element name="name" type="CharacterString"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

- IM\_ObjectIdentification 属性の集合への参照による記述例

```
<xs:complexType name="ExType">
  <xs:sequence>
    <xs:element name="name" type="CharacterString"/>
  </xs:sequence>
  <attributegroup ref="IM_ObjectIdentification"/>
</xs:complexType>
```

#### 4.6.8 共用体 (Union)

ステレオタイプが《Union》となる型は複数の属性を記述する。また、いかなる場合も属性のうちの一つだけしか存在できない。共用体は W3C XML Schema の選択構造を用いて、複合型に変換しなければならない。

**例** GM\_Position は、空間スキーマに定義された型である。これは W3C XML Schema の選択構造を用いて、ローカルの要素宣言を備えた複合型に写像される。

<<Union>> GM_Position
direct : DirectPosition indirect : GM_PointRef

図附 8-9 共用の例

```
<xs:complexType name="GM_Position">
```

```

<xs:choice>
  <xs:element name="direct" type="DirectPosition"/>
  <xs:element name="indirect" type="GM_PointRef"/>
</xs:choice>
</xs:complexType>

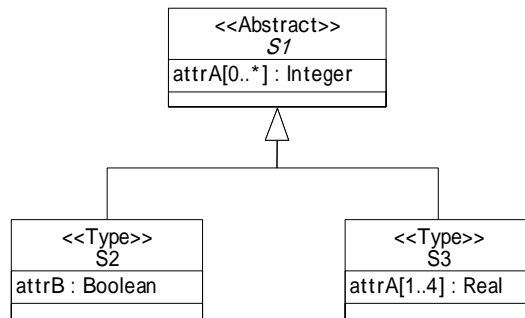
```

#### 4.7 継承 (inheritance)

UML の継承の仕組みでは、下位の型がその上位の型の属性及び関連を継承することができる。単一継承では、単一の上位の型から唯一継承が可能であるが、多重継承では二つ以上の型からの継承が可能である。UML は、単一及び多重継承の両方を認めているが、この附属書の符号化規則では、多重継承は認めないこととする。

継承は、W3C XML Schema の拡張 (extension) 又は制約 (restriction) の仕組みによって実現しなければならない。

**例** 下位型 S2, S3 を備えた上位型 S1 を定義する。S1 は抽象的な型である。



図附 8-10 継承の例

```

<xs:complexType name="S1" abstract="true">
  <xs:complexContent>
    <xs:extension base="IM_Object">
      <xs:sequence>
        <xs:element name="attrA" type="Integer" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="S2">
  <xs:complexContent>
    <xs:extension base="S1">
      <xs:sequence>
        <xs:element name="attrB" type="Boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="S3">
  <xs:complexContent>
    <xs:restriction base="S1">
      <xs:sequence>
        <xs:element name="attrA" type="Real" minOccurs="1" maxOccurs="4"/>
      </xs:sequence>
    </xs:restriction >
  </xs:complexContent>
</xs:complexType>

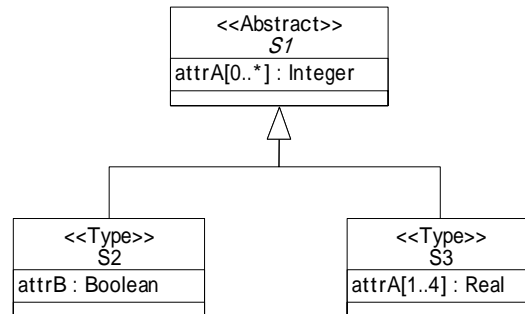
```

#### 4.7.1 代替型 (substitution types)

属性の型としての上位型の使用は、属性のインスタンスが、上位型の継承階層によって定義された具体的な下位型のうちの一つでありうることを意味する。

この符号化規則では、上位型の継承階層構造と一致する代替グループを備えたグローバルな要素を定義する。グローバルな要素は要素宣言の中で参照しなければならない。

**例** 下位型 S2, S3 を備えた上位型 S1 を定義する。S1 は抽象的な型である。



図附 8-11 代替型の例

```

<xs:element name="S1" type="S1" abstract="true"/>
<xs:element name="S2" type="S2" substitutionGroup="S1"/>
<xs:element name="S3" type="S3" substitutionGroup="S1"/>

```

#### 4.8 プロパティの要素宣言

オブジェクトのプロパティとなる属性及び関連は W3C XML Schema の要素宣言を用いて以下のように変換する。

##### 4.8.1 属性 (Attribute)

属性は型の特性を定義する。型の複合型宣言において、属性は要素宣言に変換する。

要素宣言は名称と型をもった要素を定義する。多重度は表附 8-2とする。既定時の値は表の中で示されており、宣言する必要がない。そのかわりとして、最小値と最大値が与えられる。

なお、属性が多重度 0 又は 1 をもつ基本的な型で定義されている場合、W3C XML Schema の属性宣言に変換することも可能である。

表附 8-2 要素の多重度のマッピング

UML	MinOccurs	MaxOccurs	Necessary element declaration
1 (default)	1 (default)	1 (default)	
0..1	0	1 (default)	minOccurs="0"
0..*	0	Unbounded	minOccurs="0"
1..*	1 (default)	Unbounded	minOccurs="1" maxOccurs="unbounded"
2..8	2	8	minOccurs="2" maxOccurs="8"

**例** データの型は UML で宣言される。“title”、“number”及び“subExample”という三つの属性をもっている。すべての属性は、要素宣言に割り当てる。

Example
title : CharacterString number[0..1] : Integer subExample[0..*] : Example

図附 8-12 属性の例

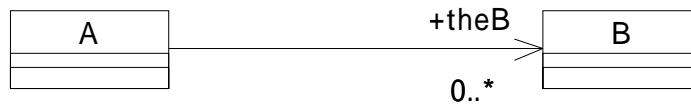
```
<xs:complexType name="Example">
  <xs:sequence>
    <xs:element name="title" type="CharacterString"/>
    <xs:element name="number" type="Integer" minOccurs="0"/>
    <xs:element name="subExample" type="Example" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

#### 4.8.2 関連 (Association)

関連は、二つの型の間一般的な関係を定義する。型のうちのひとつはソースクラスと呼び、もう一方はターゲットクラスと呼ぶ。ソースクラスのオブジェクトはターゲットクラスのオブジェクトへの参照を保持し、また逆も同様である。

関連を定義する場合は、ターゲットクラスが役割名によって特定されなければならない。ソースクラスに該当する複合型は、要素宣言を含まなければならない。要素名はターゲットクラスを特定する役割名でなければならない。その型は IM\_ObjectReference 又は基本の型でなければならないか、又は IM\_ObjectReference で定義される属性をもつ。

**例** A と B との間の関連を定義する。この関連は A から B への片方向の参照である。



図附 8-13 関連の例

```
<xs:complexType name="A">
  <xs:sequence>
    <xs:element name="theB" type="ref_B" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

ここで、ref\_B とはオブジェクト参照が可能な構造をもつ。

```
<xs:complexType name="ref_B">
  <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
```

### 4.8.3 集成 (Aggregation)

集成は、全体とそれを構成する一部分との間の弱い関係を定義する。この結びつきは弱く、同時に二つ以上の集成を構成することができる。このため、部分オブジェクトは二つ以上の集成オブジェクトによって共有してもよい。集成はこのため、その部分に対する参照を保持する。

集成型に一致する複合型は要素宣言を含まなければならない。またその宣言の名称は、部分クラスを特定する役割名に一致する。要素の型は IM\_ObjectReference に基づくものとし、ゼロ又はひとつの要素を含んでもよい。その要素は部分型に一致する。ただし、この符号化規則では、集成のインスタンスは、常に IM\_ObjectReference に基づく方法とする。

部分型が集成型を参照できる場合は、複合型である部分クラスは、要素宣言を含むものとする。要素名は、集成型を特定する要素名に一致するものとし、型は IM\_ObjectReference に基づくものとする。

**例** 集成型 C と部分型 D との間の集成関連を定義する。型 C は役割名 "theD" によって型 D を定義する。型 D は役割名 "theC" によって型 C を定義する。



図附 8-14 集成の例

```
<xs:complexType name="C">
  <xs:sequence>
    <xs:element name="theD" minOccurs="1" maxOccurs="unbounded"/>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="D" type="D" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attributeGroup ref="IM_ObjectReference"/>
  </xs:complexType>
</xs:complexType>
```

```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="D">
    <xs:sequence>
      <xs:element name="theC" type="IM_ObjectReference" minOccurs="1"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

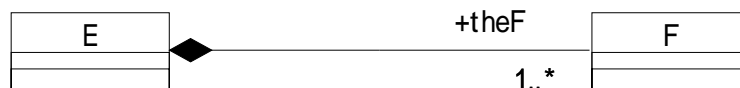
#### 4.8.4 合成 (Composition)

合成関連は、全体とそれらを構成する部分との間の強い「全体-部分」関係を定義する。合成は個々の部分を含まねばならない。

合成型に一致する複合型は、要素宣言を含まねばならない。要素宣言では、名称は部分型を特定する役割名に一致し、型は部分型の型に一致しなければならない。

部分型に一致する複合型は、要素名が合成型を特定していたとしても、要素宣言を含んではならない。部分は常に合成型に含まれるためである。

**例** E と F との間の合成を定義する。E はターゲットクラスの役割名 "theF" によって F を特定する。



図附 8-15 合成の例

```

<xs:complexType name="E">
  <xs:sequence>
    <xs:element name="theF" type="F" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

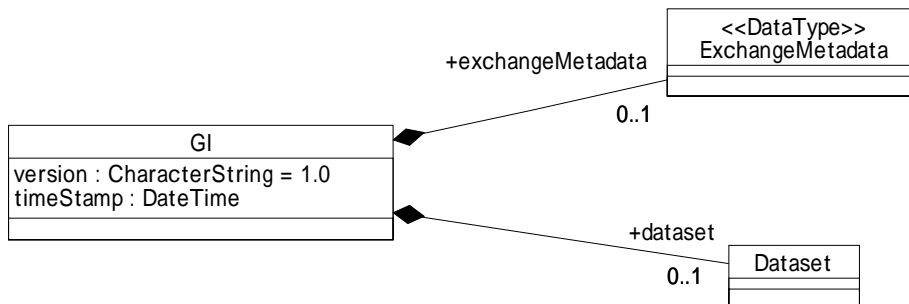
```

## 4.9 要素宣言

### 4.9.1 文書構造

交換ファイルの基底要素は GI とする。基底要素 GI は、交換ファイルのメタデータである ExchangeMetadata 及びデータ集合である Dataset の二つの要素をもつことができる。





図附 8-16 文書構造

GI は、二つの属性をもつことができる。属性 version は、交換ファイルが参照する JPGIS の版を示す。属性 timeStamp はデータを符号化した日時を示す。GI の宣言を以下に示す。

```

<xs:element name="GI">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="exchangeMetadata" type="ExchangeMetadata"
        minOccurs="0"/>
      <xs:element name="dataset" type="Dataset" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="version" type="CharacterString" use="required"
      fixed="1.0"/>
    <xs:attribute name="timeStamp" type="DateTime" use="required"/>
  </xs:complexType>
</xs:element>
  
```

#### 4.9.2 データ集合とオブジェクト

データ集合 (Dataset) は一つ以上のオブジェクトをもつ。ここでオブジェクトとは、応用スキーマに定義された型のうち、ステレオタイプが《DataType》, 《Type》, 《Feature》及び《Abstract》となる型である。オブジェクトは、要素宣言をする必要がある。また、データ集合は、参照系への参照をもつことができる。



図附 8-17 データ集合とオブジェクト

データ集合の宣言を以下に示す。以下の例では、このデータ集合は、” namespace” で指定した名前空間に含まれる Object を要素としてもつことができる。

```

<xs:element name="Dataset" type="Dataset"/>
<xs:complexType name="Dataset">
  
```

```

<xs:sequence maxOccurs="unbounded">
  <xs:group ref="Object"/>
</xs:sequence>
<xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

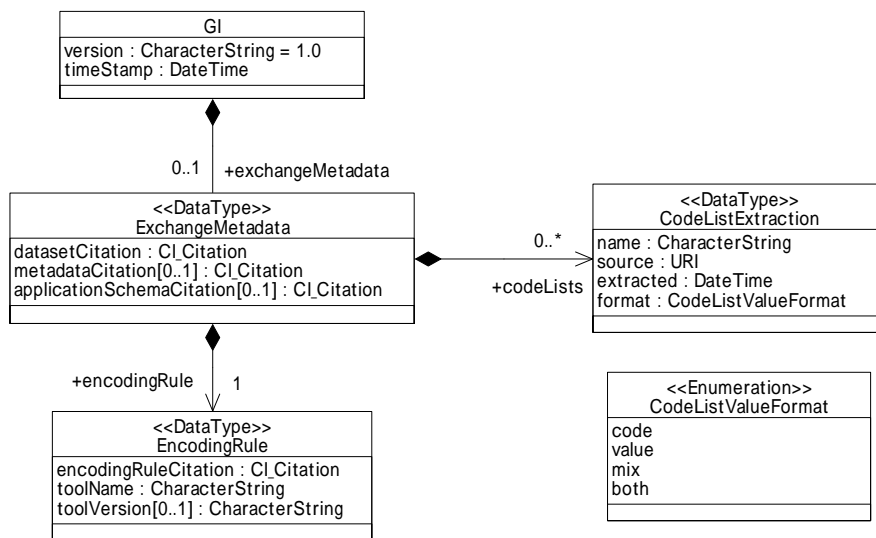
<xs:group name="Object">
  <xs:sequence>
    <xs:any namespace="Object が含まれる名前空間"/>
  </xs:sequence>
</xs:group>

```

### 4.9.3 交換メタデータ

交換メタデータ (ExchangeMetadata) の型を図附 8-19 に示す。CI\_Citation 型及び URI 型は、附属書 6 で規定している。

GI は、役割名 exchangeMetadata によって、ExchangeMetadata をもつことができる。



図附 8-18 基底要素と交換用メタデータ

ExchangeMetadata は、データ集合を記述する情報をもたなければならない。

属性 datasetCitation は、データ集合の起源を記述する。metadataCitation 属性には、メタデータの起源を記述してよい。また applicationSchemaCitation 属性には、応用スキーマの起源を記述してよい。

役割名 encodingRule によって、符号化規則を記述しなくてはならない。またデータ集合が符号リストの属性をもつならば、役割名 codeLists によって符号リストとその値を示さなければならない。

役割名 encodingRule によって合成される EncodingRule は、データ集合の生成に摘要した符号化規則についての情報を記述する。属性 encodingRuleCitation は、摘要した符号化規則の引用を記述し、属性 toolName にはデータ集合を生成するのに用いた符号化ツールを記述し、属性 toolVersion には、そのツールの版を記述してよい。

CodeListExtraction は、name 属性及び source 属性によってそのコードを識別する。またこのコードがいつ引用されたかを extracted 属性に記述する。format 属性には、コードを用いるのか、コードに対応する値をとるのか、あるいは混合して使用するのか、両方とも使用するのかを記載する。

ExchangeMetadata の宣言を以下に示す。

```
<xs:element name="ExchangeMetadata" type="ExchangeMetadata"/>
<xs:complexType name="ExchangeMetadata">
  <xs:sequence>
    <xs:element name="datasetCitation" type="CI_Citation"/>
    <xs:element name="metadataCitation" type="CI_Citation" minOccurs="0"/>
    <xs:element name="applicationSchemaCitation" type="CI_Citation"
      minOccurs="0"/>
    <xs:element name="encodingRule" type="EncodingRule"/>
    <xs:element name="codeLists" type="CodeListExtraction" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

EncodingRule の宣言を以下に示す。

```
<xs:complexType name="EncodingRule">
  <xs:sequence>
    <xs:element name="encodingRuleCitation" type="CI_Citation"/>
    <xs:element name="toolName" type="CharacterString"/>
    <xs:element name="toolVersion" type="CharacterString" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

CodeListExtraction の宣言を以下に示す。

```
<xs:element name="CodeListExtraction" type="CodeListExtraction"/>
<xs:complexType name=" CodeListExtraction " >
  <xs:sequence>
    <xs:element name="name" type="CharacterString"/>
    <xs:element name="source" type="URI"/>
    <xs:element name="extracted" type="DateTime"/>
    <xs:element name="format" type="CodeListValueFormat"/>
  </xs:sequence>
</xs:complexType>
```

CodeListValueFormat の宣言を以下に示す。

```
<xs:simpleType name=" CodeListValueFormat " >
  <xs:restriction base="xs:string">
    <xs:enumeration value="code"/>
    <xs:enumeration value="value"/>
    <xs:enumeration value="mix"/>
    <xs:enumeration value="both"/>
  </xs:restriction >
</xs:simpleType >
```

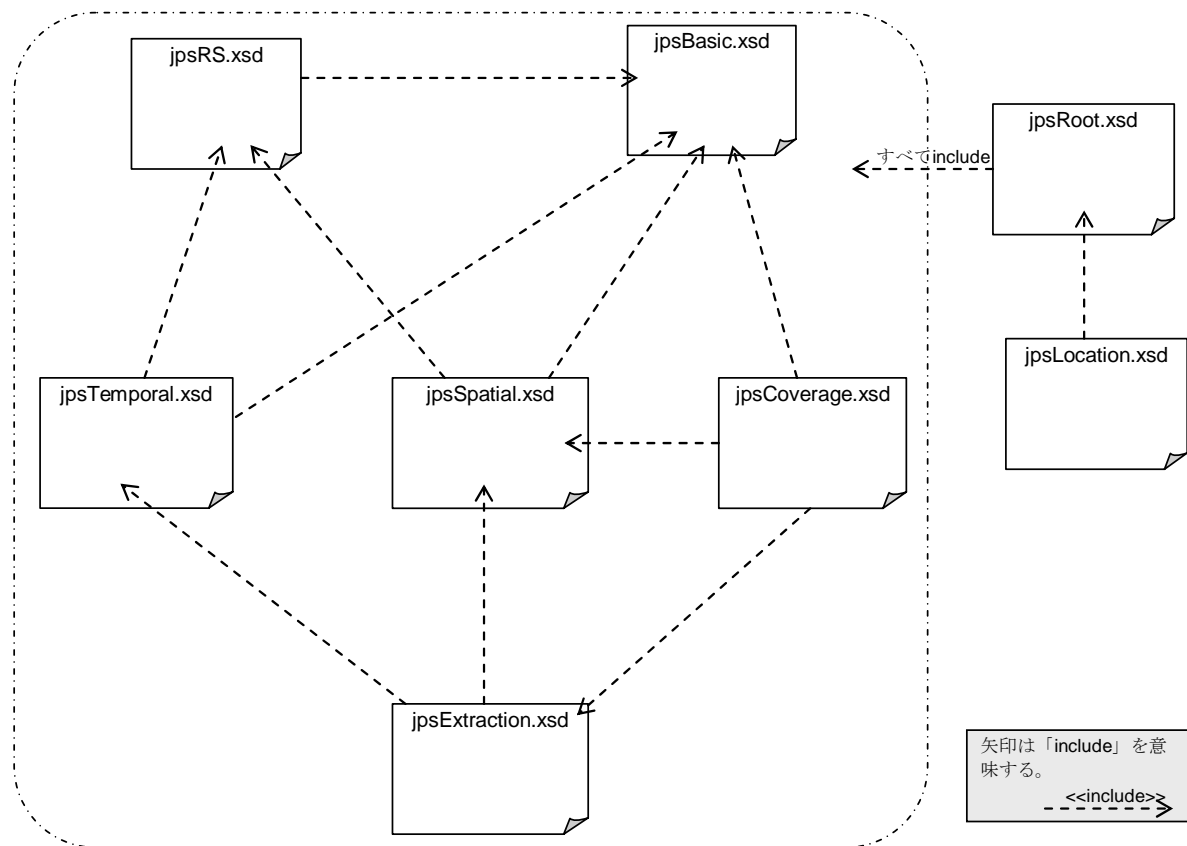
## 5. JPGIS 標準スキーマの XML Schema

### 5.1 序文

ここでは、この附属書の符号化規則に則った JPGIS 標準スキーマの W3C XML Schema (以下、標準 XML Schema) を示す。標準 XML Schema には、基本データ型、空間スキーマ、時間スキーマ、被覆の幾何及び関数のためのスキーマ、地理識別子による空間参照スキーマ、参照系スキーマ及び他の規格から引用するクラスのスキーマが含まれる。また、地物カタログ化法の W3C XML Schema をテンプレート (以下、標準 XML Schema テンプレート) として示す。

標準 XML Schema のタグ命名規則は、地物は「地物名」、属性は「属性名」、関連は「関連名」としている。応用スキーマの W3C XML Schema についても基本的にこの規則に従うこととする。

### 5.2 標準 XML Schema の構成と概要



図附 8-19 標準 XML Schema の構成

#### 5.2.1 ファイル構成

標準 XML Schema は以下のファイルから構成される。

1. jpsBasic.xsd :  
JPGIS 「5 基本データ型」 で規定する基本データ型の W3C XML Schema を定義する。
2. jpsSpatial.xsd :

JPGIS「7 空間スキーマ」で規定する空間スキーマの W3C XML Schema を定義する。jpsSpatial.xsd は jpsBasic.xsd 及び jpsRS.xsd で定義する要素を参照するため、これらのファイルを include する。

3. jpsTemporal.xsd :

JPGIS「8 時間スキーマ」で規定する時間スキーマの W3C XML Schema を定義する。jpsTemporal.xsd は jpsBasic.xsd で定義する要素を参照するため、jpsBasic.xsd を include する。

4. jpsCoverage.xsd :

JPGIS「9 被覆の幾何及び関数のためのスキーマ」で規定する被覆スキーマの W3C XML Schema を定義する。jpsCoverage.xsd は、jpsBasic.xsd, jpsSpatial.xsd, jpsRS.xsd 及び jpsExtraction.xsd で定義する要素を参照するため、これらのファイルを include する。

5. jpsLocation.xsd :

JPGIS「10 地理識別子による空間参照」で規定される地理識別子スキーマの W3C XML Schema を定義する。また、JPGIS「附属書 2 (規定) 参照系」で規定される参照系スキーマのうち、RS\_LRS の W3C XML Schema もここで定義する。jpsLocation.xsd は、jpsRoot.xsd で定義する要素を参照するため、このファイルを include する。

6. jpsRS.xsd :

JPGIS「附属書 2 (規定) 参照系」で規定される参照系スキーマの W3C XML Schema を定義する。ただし、RS\_LRS については、jpsLocation.xsd に定義する。jpsRS.xsd は、jpsBasic.xsd で定義する要素を参照するため、このファイルを include する。

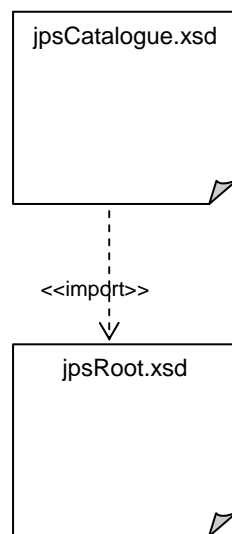
7. jpsExtraction.xsd :

JPGIS「附属書 6 (規定) 他の規格から引用するクラス」で規定されるクラスの W3C XML Schema を定義する。jpsExtraction.xsd は、jpsBasic.xsd, jpsSpatial.xsd 及び jpsTemporal.xsd で定義する要素を参照するため、これらのファイルを include する。

8. jpsRoot.xsd :

標準 XML Schema のルートとなる W3C XML Schema を定義する。jpsRoot.xsd は、1 から 7 の全ての標準 XML Schema を include する。また、交換メタデータのスキーマの W3C XML Schema を定義する。

### 5.3 標準 XML Schema テンプレートの構成と概要



図附 8-20 標準 XML Schema の構成

### 5.3.1 ファイル構成

標準 XML Schema テンプレートは以下のファイルから構成される。使用者は、このテンプレートをを用いて、XML 文書を作成することができる。

#### 1. jpsCatalogue.xsd :

JPGIS「11 地物カタログ化法」で規定する地物カタログスキーマの W3C XML Schema を定義する。jpsCatalogue.xsd は jpsRoot.xsd で定義する要素を参照するため、jpsRoot.xsd を import する。

### 5.4 名前空間

標準 XML Schema の名前空間及び名前空間接頭辞は以下の通りとする。

#### 1. 名前空間

[http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0\\_2008-04](http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0_2008-04)

#### 2. 名前空間接頭辞

jps

### 5.5 標準 XML Schema と応用スキーマの XML Schema

JPGIS に則って作成される応用スキーマの符号化規則が W3C XML Schema で示され、かつ、JPGIS で規定する標準 XML Schema を使用する場合は、以下の規則に従わなければならない。

1. 標準 XML Schema の xsd ファイルとは別に応用スキーマの W3C XML Schema の xsd ファイルを作成する。
2. 応用スキーマの W3C XML Schema が標準 XML Schema を利用する場合は、標準 XML Schema を import する。
3. 応用スキーマの W3C XML Schema で定義する名前空間及び名前空間接頭辞は、標準 XML Schema で規定する名前空間及び名前空間接頭辞とは異なるものとする。
4. 応用スキーマの W3C XML Schema のタグ名には、標準 XML Schema で使用されているタグ名と同じタグ名を付けてはならない。

### 5.6 標準 XML Schema の拡張及び制限のための規則

#### 5.6.1 一般

標準 XML Schema を拡張及び制限する場合の規則を定める。

#### 5.6.2 拡張のための規則

標準 XML Schema を拡張して用いる場合は、以下の規則に従わなければならない。

##### 規則

1. 標準 XML Schema に規定していない基本データ型、空間要素、時間要素、被覆要素の XML Schema を定義する場合は、標準 XML Schema の xsd ファイルとは別に xsd ファイルを作成し、定義しなければならない。
2. 拡張標準 XML Schema として作成した xsd ファイルは、標準 XML Schema を import して使用しなければならない。
3. 拡張標準 XML Schema を使用する応用スキーマの XML Schema は、拡張標準 XML Schema を import して使用しなければならない。

### 5.6.3 制限のための規則

標準 XML Schema を制限し、その一部を使用することができる。

#### 規則

1. 標準 XML Schema に規定された基本データ型、空間要素、時間要素及び被覆要素を制限して使用する場合は、標準 XML Schema の xsd ファイルとは別に xsd ファイルを作成し、定義しなければならない（制限標準 XML Schema）。
2. 制限標準 XML Schema として作成した xsd ファイルは、標準 XML Schema を import して使用しなければならない。
3. 制限標準 XML Schema を使用する応用スキーマの XML Schema は、制限標準 XML Schema を import して使用しなければならない。

### 5.7 標準 XML Schema 文書

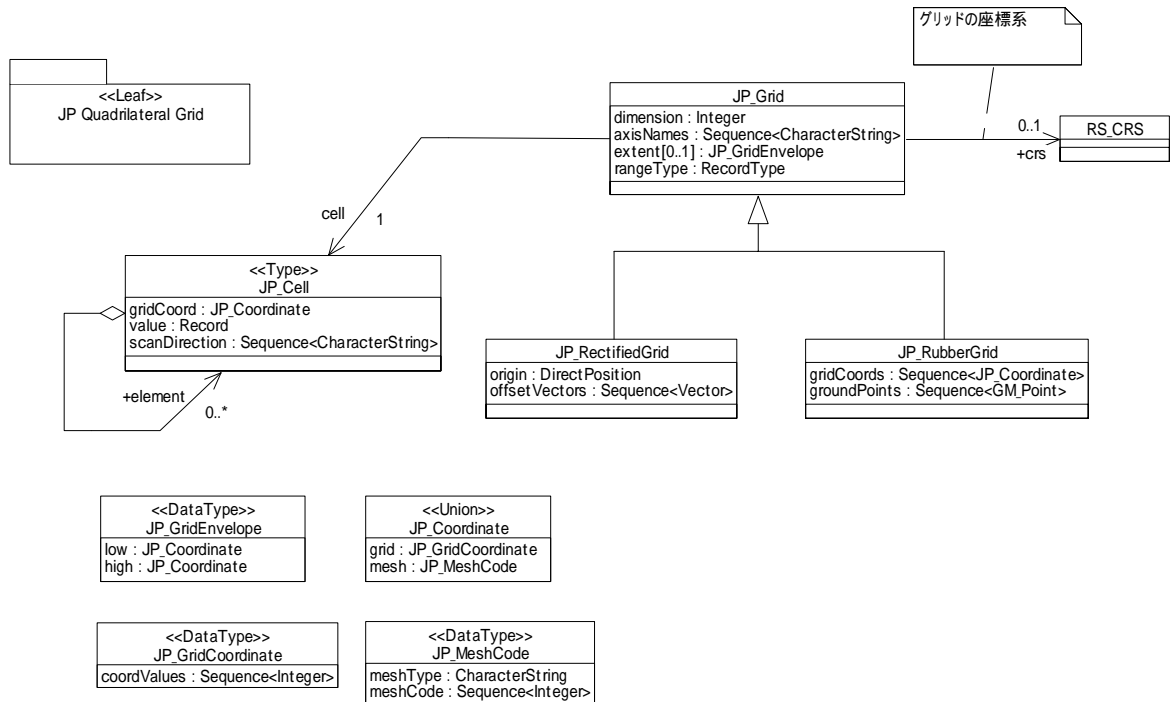
標準 XML Schema は次の URL のファイルを使用する。

[http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0\\_2008-04](http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0_2008-04)

## 附属書 9 (参考) 四辺形グリッド被覆

**序文** この附属書では、四辺形グリッド被覆のための標準的なスキーマの定義を明示する。この附属書は、ISO 19123 (Geographic information - Schema for coverage geometry and functions) を参考にし、JPGIS 独自に定義したものであり、規定の一部ではない。

### 1. 四辺形グリッド被覆パッケージ



図附 9-1 四辺形グリッド被覆パッケージ

グリッド被覆は、空間定義域の系統的なモザイク化を採用している。このようなモザイク化の主な利点は、空間定義域の要素の連続的な列挙を支持することで、データの保存とアクセスを効率化できるという点にある。

JPGIS では、日本において一般に採用されているグリッドデータ又はメッシュデータの構造を支持するために、JPGIS 独自にこの附属書を示す。

#### 1.1 JP\_Grid

JP\_Grid (図附 9-1) は、四辺形グリッドの概要を示す。

##### 属性

1. dimension:Integer は、グリッドの次元を示す。
2. axisNames:Sequence<CharacterString>は、グリッドの軸の名前を記述する。
3. extent[0..1]:JP\_GridEnvelope は、任意の属性で、グリッドの領域の境界を示す。
4. rangeType:RecordType は、被覆の値域を記述する。データ型 RecordType は附属書 6 で規定しており、属性名とデータ型の対のリストで構成する。



## 関連

1. cell 関連役割は、JP\_Grid を、グリッド被覆の最も大きい単位となるグリッドセル (JP\_Cell) と関連付ける。

**備考 1** JP\_Cell は階層構造をもつことができ、cell 関連役割は、JP\_Cell の階層構造の最上位の JP\_Cell を参照する。

**備考 2** 四辺形グリッド被覆がメッシュの場合、JP\_Cell はメッシュと同義となる。

2. crs 関連役割は、JP\_Grid を、このグリッド被覆に基づくグリッド座標系と関連付ける。

### 1.2 JP\_RectifiedGrid

JP\_RectifiedGrid (図附 9-1) は、グリッド座標系における JP\_Grid を、外部座標参照系における原点と、向きと大きさをもつベクトルによって幾何補正するグリッド被覆を示す。

#### 属性

1. [上位型から継承する属性]dimension:Integer は、グリッドの次元を示す。
2. [上位型から継承する属性]axisNames:Sequence<CharacterString>は、グリッドの軸の名前を記述する。
3. [上位型から継承する属性]extent[0..1]:JP\_GridEnvelope は、任意の属性で、グリッドの領域の境界を示す。
4. [上位型から継承する属性]rangeType:RecordType は、被覆の値域を記述する。データ型 RecordType は附属書 6 で規定しており、属性名とデータ型の対のリストで構成する。
5. origin:DirectPosition は、外部座標参照系における補正グリッド被覆の原点を示す。データ型 DirectPosition は、7 空間スキーマで規定する。
6. offsetVectors:Sequence<Vector>は、外部座標参照系におけるグリッドの各軸方向における割当を決定するベクトルの列を示す。データ型 Vector は、5 基本データ型で規定する。

**備考** offsetVectors のベクトルの数は、グリッド被覆の次元よりも小さくなければならない。

#### 関連

1. [上位型から継承する関連役割]cell 関連役割は、JP\_Grid を、グリッド被覆の最も大きい単位となるグリッドセル (JP\_Cell) と関連付ける。

**備考 1** JP\_Cell は階層構造をもつことができ、cell 関連役割は、JP\_Cell の階層構造の最上位の JP\_Cell を参照する。

**備考 2** 四辺形グリッド被覆がメッシュの場合、JP\_Cell はメッシュと同義となる。

2. [上位型から継承する関連役割]crs 関連役割は、JP\_Grid を、このグリッド被覆に基づくグリッド座標系と関連付ける。

### 1.3 JP\_RubberGrid

JP\_RubberGrid (図附 9-1) は、グリッド座標系における JP\_Grid を、グリッド座標値一つ一つを外部座標参照系における座標と対応させることによって座標変換を行うグリッド被覆を示す。

#### 属性

1. [上位型から継承する属性]dimension:Integer は、グリッドの次元を示す。
2. [上位型から継承する属性]axisNames:Sequence<CharacterString>は、グリッドの軸の名前を記述する。
3. [上位型から継承する属性]extent[0..1]:JP\_GridEnvelope は、任意の属性で、グリッドの領域の境界を示す。
4. [上位型から継承する属性]rangeType:RecordType は、被覆の値域を記述する。データ型 RecordType は附属書 6 で規定しており、属性名とデータ型の対のリストで構成する。

5. `gridCoords:Sequence<JP_Coordinate>`は、グリッド座標系における `JP_Cell` のグリッド座標値の列を示す。
6. `groundPoints:Sequence<GM_Point>`は、`JP_RubberGrid` の属性 `gridCoords` の `JP_Coordinate` の列と対応する外部座標参照系における `GM_Point` の列を示す。データ型 `GM_Point` は、7 空間スキーマで規定する。

#### 関連

1. [上位型から継承する関連役割] `cell` 関連役割は、`JP_Grid` を、グリッド被覆の最も大きい単位となるグリッドセル (`JP_Cell`) と関連付ける。
  - 備考 1** `JP_Cell` は階層構造をもつことができ、`cell` 関連役割は、`JP_Cell` の階層構造の最上位の `JP_Cell` を参照する。
  - 備考 2** 四辺形グリッド被覆がメッシュの場合、`JP_Cell` はメッシュと同義となる。
2. [上位型から継承する関連役割] `crs` 関連役割は、`JP_Grid` を、このグリッド被覆に基づくグリッド座標系と関連付ける。

#### 1.4 JP\_GridEnvelope

`JP_GridEnvelope` は、`JP_Grid` の反対側に向かい合った隅のグリッド座標値 (`JP_Coordinate` (附属書 9 の 1.6) ) を示すデータ型である。

#### 属性

1. `low:JP_Coordinate` は、`JP_Grid` に含まれるすべてのグリッド点のうち、最小のグリッド座標値である。
2. `high:JP_Coordinate` は、`JP_Grid` に含まれるすべてのグリッド点のうち、最大のグリッド座標値である。

#### 1.5 JP\_Cell

`JP_Cell` は、グリッド被覆を構成するグリッドセルを表す。

#### 属性

1. `gridCoord:JP_Coordinate` は、`JP_Cell` の四隅の中で最小のグリッド座標値である。
  - 備考** `gridCoord` は、グリッドセルの代表点を示す。
2. `value:Record` は、グリッドセルがもつ地物属性値レコードを示す。
3. `scanDirection:Sequence<CharacterString>`は、軸名のリストである。該当する `JP_Cell` の一つ下の階層の `JP_Cell` の配置順序を示す。ただし、ここで指定する軸名は、`JP_Grid` の属性 `axisNames` で定義した軸名を使用しなければならない。

**備考** `JPGIS` では、`JP_Cell` の配置は線形とする。

#### 関連

1. `element` 関連役割は、`JP_Cell` を、一つ下の階層の `JP_Cell` の集合と関連付ける。`JP_Cell` が最下層の場合は、この関連役割は"0"となる。

#### 1.6 JP\_Coordinate

`JP_Coordinate` は、`JP_GridCoordinate` 又は `JP_MeshCode` への参照のどちらかからなる共用体である。

#### 属性

1. 属性 `grid:JP_GridCoordinate` は、グリッド被覆の座標を示す `JP_GridCoordinate` への参照である。

2. 属性 `mesh:JP_MeshCode` は、日本において使用されるメッシュデータのメッシュコードを示す `JP_MeshCode` への参照である。

### 1.7 JP\_GridCoordinate

`JP_GridCoordinate` は、グリッド座標をもつためのデータ型である。

#### 属性

1. 属性 `coordValues:Sequence<Integer>` は、グリッドの各次元に対応した整数値の列である。これらの座標値の順序付けは、`JP_Grid` の `axisNames` 属性の要素の順序付けと同じでなければならない。単一座標の値は、特定の軸の方向でグリッドの原点から伸びているオフセットの数と同じになる。

### 1.8 JP\_MeshCode

`JP_MeshCode` は、日本において整備されているメッシュデータのメッシュコードを示すデータ型である。

**例** メッシュデータの代表的なものとして、地域メッシュがある。地域メッシュをこのプロファイルを使用して定義する場合、`JP_MeshCode` は JIS X 0410 に規定されているメッシュコードとなる。

#### 属性

1. `meshType:CharacterString` は、定義するメッシュデータの名称を示す。

**例** 地域メッシュ

2. `meshCode:Sequence<Integer>` は、メッシュコードを示す。メッシュデータが階層構造を持ち、かつそのメッシュコードの命名規則に階層がある場合は、メッシュコードを、順序を持った列で記述することができる。

**例** 地域メッシュの三次メッシュコードを記述する場合は、「一次区画を示す数字、二次区画を示す数字、三次区画を示す数字」の順で記述する。

## 2. 四辺形グリッド被覆 XML Schema 文書

四辺形グリッド被覆の XML Schema 文書は次の URL にあるファイル `jpsGrid.xsd` を使用する。

[http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0\\_2008-04](http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0_2008-04)

## 附属書 10 (参考) 描画法

**序文** この附属書は、ISO 19117 (Geographic information - Portrayal) を参考にし、地理情報標準プロファイルにおける描画法の基本的な考え方を示すとともに空間データ製品仕様書に記載すべき地物の描画規則 (描画カタログ) を示すものであり、規定の一部ではない。

### 1. 用語の定義

#### 1.1 地物描画

地物を表示すること。紙地図のみならず、ディスプレイ上での表示も含む。

#### 1.2 描画辞書

地物の描画に必要なデータのうち、共通利用が可能なものを辞書として定義するもの。

#### 1.3 描画辞書スキーマ

描画辞書の構造を記述したもの。UML クラス図と定義文書からなる。

#### 1.4 描画辞書インスタンス

描画辞書スキーマに従って作成されたインスタンス。

#### 1.5 地物描画スキーマ

地物の描画に必要なデータのうち、個々の地物インスタンスがもつべきデータの構造を記述したもの。UML クラス図と定義文書からなる。

#### 1.6 地物描画インスタンス

地物描画スキーマに従って作成された個々の地物インスタンスに必要なデータ。

#### 1.7 地図記号辞書

地図記号を収録した辞書。

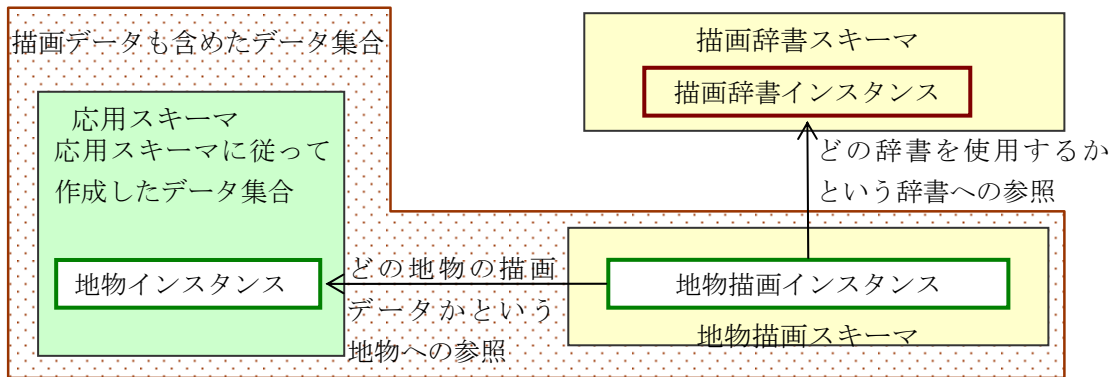
#### 1.8 注記スタイル辞書

注記の表記法 (スタイル) を収録した辞書。

### 2. 描画法の基本的な考え方

#### 2.1 地物描画インスタンスと描画辞書

地物描画に必要なデータには、地図記号を表示する位置や注記の文字列のように個々の地物インスタンスに固有のデータ、それから、地図記号そのものや注記の表示方法など個々の地物インスタンスが共通利用できるデータの二種類がある。本附属書では前者を地物描画インスタンス、後者を描画辞書インスタンスと呼ぶ。地物描画インスタンスは、どの地物の描画データであるかという地物インスタンスへの参照と、どの描画辞書インスタンスを使用するかという辞書への参照をもつ。空間データの作成者は、応用スキーマに従ったデータ集合のみを作成する場合と、地物描画インスタンスを含めたデータ集合を作成する場合がある。

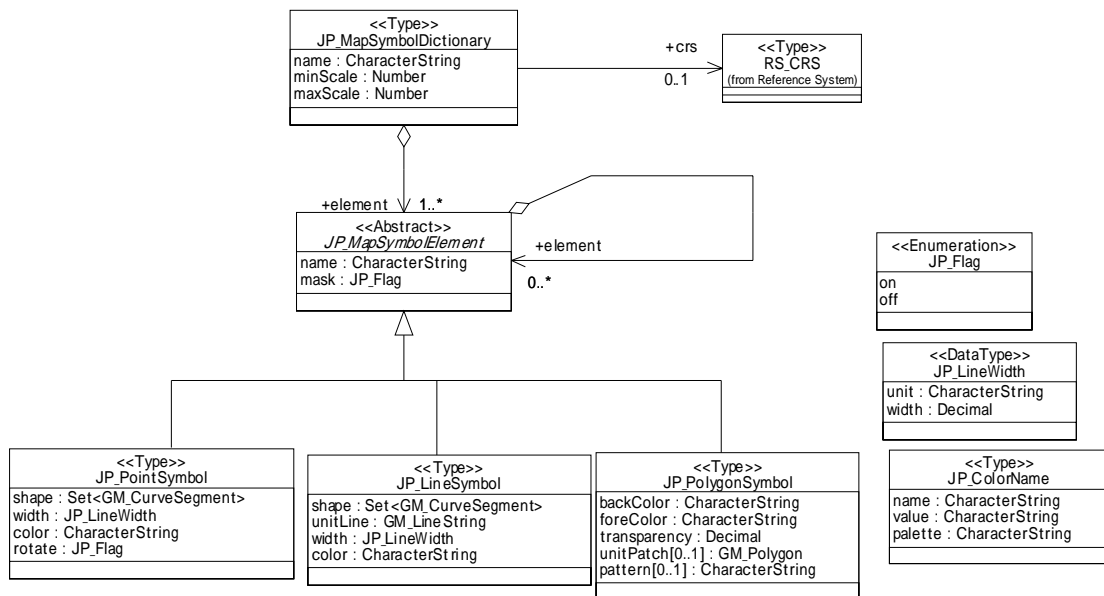


図附 10-1 描画データと地物インスタンスの関係

### 3. 描画辞書スキーマ

描画辞書とは、地図記号そのものや注記の表示方法など、個々の地物インスタンスが共通利用できるデータである。個々の地物インスタンスが共通利用できるデータは、データ集合として用意しておき、各地物インスタンスが共通利用することによって、データの冗長性等を防ぐことができる。また、異なるデータ作成者やデータ利用者が、地図記号や注記のスタイルが格納された同じデータ集合を用いることによって、いつでも誰でも同じ地図を表示することが可能となる。本附属書では、地図記号のための地図記号辞書、及び注記のための注記スタイル辞書の二種類の辞書のスキーマを定める。

#### 3.1 地図記号辞書スキーマ



図附 10-2 JP\_MapSymbolDictionary クラス図

### 3.1.1 JP\_MapSymbolDictionary

地図記号の形状や色、線種等を定めた、地図記号のための辞書である。

#### 属性

1. name:CharacterString は、この地図記号辞書の識別子である。
2. maxScale : Number は、この地図記号辞書が適用される最大縮尺を記載する。
3. minScale : Number は、この地図記号辞書が適用される最小縮尺を記載する。

#### 関連

1. 役割名 element によって、この地図記号辞書を構成する地図記号要素を集成する。
2. 役割名 crs によって、この地図記号辞書に集成された地図記号要素の形状を定義するための座標参照系と関連する。初期値は原点 (0, 0) , 単位を mm とし、右方向を X 上方向を Y とする XY 直行座標系とする。初期値をとる場合には、座標参照系への参照を記述しなくて良い。

### 3.1.2 JP\_MapSymbolElement

地図記号辞書を構成する要素を示す抽象型。

#### 属性

1. name:characterString は、この地図記号の識別子である。
2. mask : JP\_Flag は、地図記号要素が、他の地図記号又は注記と重なった際に、マスク表示を行うか否かの別である。マスク表示を行う場合は on, 行わない場合は off とする。初期値は on とする。

#### 関連

1. 役割名 element によって、一つの地図記号要素を下位の型である点記号や線記号及び面記号の集まりを表現する。

### 3.1.3 JP\_PointSymbol

点状の地図記号要素。建物の用途を示す記号や、記念碑や樹木など単独で存在する地物を描画するために用いることができる。

#### 属性

1. [上位型から継承する属性]name:CharacterString は、この地図記号の識別子である。
2. [上位型から継承する属性]mask : JP\_Flag は、地図記号要素が、他の地図記号又は注記と重なった際に、マスク表示を行うか否かの別である。マスク表示を行う場合は on, 行わない場合は off とする。初期値は on とする。
3. shape : Set<GM\_CurveSegment>は、この点記号を構成する曲線分の集まりである。
4. width : JP\_LineWidth は、点記号の形状を表示する際の線の幅である。ただし、この点記号を集成する地図記号辞書に定義された対象縮尺で表示した際の線の幅とする。
5. color : CharacterString は、点記号要素を構成する曲線分の色名である。この色名は、ColorName 型の属性 name の値と一致しなくてはならない。
6. rotate : JP\_Flag は、点記号を、表示範囲の回転によらず、一定の角度を保ってもよいかどうかの別である。on の場合は、角度を一定に保つことが可能（絶対座標に対して可変）, off の場合は、表示範囲の回転角度に応じて、点記号の角度を変更しなくてはならない（絶対座標に対して固定）。初期値は on とする

**例** 流水方向などの記号は角度が意味をもつため、表示範囲を回転させたときにその回転方向とともに記号を回転させることは意味をなさない。この場合、属性 回転の値は off とする。

#### 関連

1. [上位型から継承する関連]役割名 element によって、点記号を構成する他の地図記号要素を集成する。

### 3.1.4 JP\_LineSymbol

線状の地図記号要素。行政界や植生界など場所を示す地物を描画するために用いることができる。

#### 属性

1. [上位型から継承する属性]name:CharacterString は、この地図記号の識別子である。
2. [上位型から継承する属性]mask:JP\_Flag は、地図記号要素が、他の地図記号又は注記と重なった際に、マスク表示を行うか否かの別である。マスク表示を行う場合は on, 行わない場合は off とする。初期値は on とする。
3. shape : Set<GM\_CurveSegment>は、この線記号を構成する曲線分の集まりである。
4. unitLine:GM\_LineString は、この線記号の繰り返しの単位となる線分である。始点と終点から構成する。属性 shape で構成される曲線分の集まりは、属性 unitLine で示される場所に含まれなくてはならない。
5. width:LineWidth は、線記号の形状を表示する際の線の幅である。ただし、この線記号を集成する地図記号辞書に定義された対象縮尺で表示した際の線の幅とする。
6. color:CharacterString は、線記号を構成する曲線分の色名である。この色名は、ColorName 型の属性 name の値と一致しなくてはならない。

#### 関連

1. [上位型から継承する関連]役割名 element によって、線記号を構成する他の地図記号要素を集成する。

### 3.1.5 JP\_PolygonSymbol

土地の被覆や用途など面の広がりをもつ地物の描画に適用することができる。また、地図記号を構成する要素としても使用することができる。

#### 属性

1. [上位型から継承する属性]name:CharacterString は、この地図記号の識別子である。
2. [上位型から継承する属性]mask : JP\_Flag は、地図記号要素が、他の地図記号又は注記と重なった際に、マスク表示を行うか否かの別である。マスク表示を行う場合は on, 行わない場合は off とする。初期値は on とする。
3. backColor:CharacterString は面の塗りつぶしの背景となる色の名称である。この色名は、ColorName 型の属性 name の値と一致しなくてはならない。
4. foreColor:CharacterString は面の塗りつぶしの前景となる色の名称である。この色名は、ColorName 型の属性 name の値と一致しなくてはならない。
5. transparency:Decimal は、透過率を示す 0 から 1 までの値である。0 の場合は不透過であり、1 を透過とする。
6. unitPatch[0..1]:GM\_Polygon は、面に前景色を用いた模様を記述する場合に、繰り返しの単位となる矩形である。この unitPatch で定義された矩形の範囲内に役割名 element で関連付けられた他の地図記号要素を配置することにより、面の模様を作成する。
7. pattern[0..1]:CharacterString は、面に前景色を用いた模様を記述する場合に、他で定義された模様を用いる場合にその識別子を記述することにより、面の模様を定義する。

#### 関連

1. [上位型から継承する関連]役割名 element によって、面記号の模様を構成する他の地図記号要素を集成する。

### 3.1.6 JP\_LineWidth

描画時の線の幅を定義する。

#### 属性

1. unit:CharacterString は、線号やピクセルなど線の幅の単位を示す。
2. width:Decimal は、属性 unit で定義された単位で記載された線の幅を示す。ただし、この線の幅はこれが用いられる地図記号要素が収録される地図記号辞書に定義された対象縮尺で表示した際の線の幅とする。

### 3.1.7 JP\_ColorName

描画時の地図記号要素に用いる色を定義する。

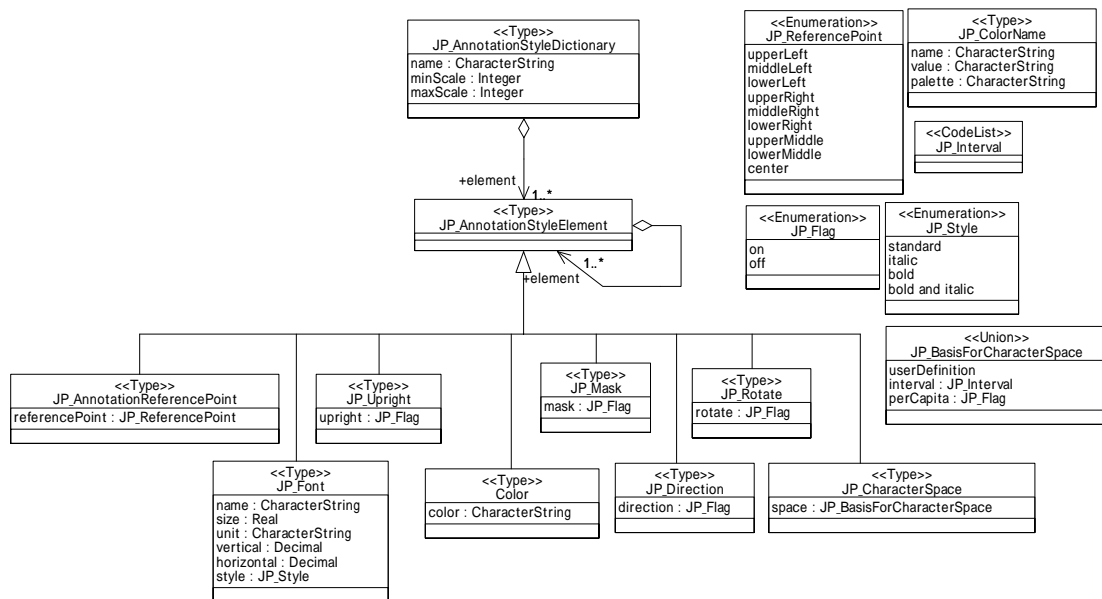
#### 属性

1. name:CharacterString は、この色を識別するための名称である。
2. value:CharacterString は、属性 palette で定義された色の記述方法に基づき記述された色を構成する値である。
3. palette:CharacterString は、RGB や SVCN など色の記述方法の名称である。

### 3.1.8 JP\_Flag

属性の値を記述する列挙型である。「on」または「off」のいずれかの値をとる。

## 3.2 注記スタイル辞書スキーマ



図附 10-3 JP\_AnnotationStyleDictionary クラス図

### 3.2.1 JP\_AnnotationStyleDictionary

注記の表記法（スタイル）のための辞書。

#### 属性

1. name:characterString は、この注記スタイル辞書の識別子である。
2. maxScale: Number は、この注記スタイル辞書が適用される最大縮尺を記載する。



3. minScale : Number は、この注記スタイル辞書が適用される最小縮尺を記載する。

#### 関連

1. 役割名 element によって、この注記スタイル辞書を構成する注記表示要素を集成する。

### 3.2.2 JP\_AnnotationStyleElement

注記の表記法（スタイル）を構成する要素。

#### 関連

1. 役割名 element は、注記表示要素を構成する注記表示要素の下位クラスを集成する。自身への参照は含まない。また、下位クラスは継承しない。この関連により、いくつかの注記表示要素を集合したものを一つのインスタンスとして扱うことができる。

### 3.2.3 JP\_AnnotationReferencePoint

注記文字列を配置する際に参照する点。JP\_AnnotationStyleElement を継承する。

#### 属性

1. referencePoint:JP\_ReferencePoint は、文字列を表示する際に参照する点である。この属性の型である JP\_ReferencePoint は upperLeft(左上), middleLeft(左中央), lowerLeft(左下), upperRight(右上), middleRight(右中央), lowerRight(右下), upperMiddle(上中央), lowerMiddle(下中央), center(中央)を値とする列挙型であるが、任意に拡張または制限してよい。

### 3.2.4 JP\_Color

注記文字列の色。JP\_AnnotationStyleElement を継承する。

#### 属性

1. color:CharacterString は、注記文字列の色の名前である。ColorName 型の属性 name の値と一致しなくてはならない。

### 3.2.5 JP\_Font

注記文字列のフォント。JP\_AnnotationStyleElement を継承する。

#### 属性

1. name:CharacterString は、フォントの識別子。
2. size:Decimal は、文字の大きさ。属性 unit に示された単位を用いて対象縮尺で表示したときの大きさとする。
3. unit:CharacterString は、mm やピクセルなど文字の大きさを記述する単位である。
4. verticalRatio:Decimal は、文字列の高さの拡大・縮小である。サイズで指定された高さに対する比率によって表す（単位：%）。
5. horizontalRatio:Decimal は、文字列の高さの拡大・縮小である。サイズで指定された幅に対する比率によって表す（単位：%）。
6. style:JP\_Style は、文字列の表示スタイルを示す。型であるスタイルは、standard(標準), italic(斜体), bold(太字), bold and italic(太字斜体)からなる列挙型であるが、任意に拡張または制限してよい。

### 3.2.6 JP\_Characterspace

注記文字列の文字間隔（字隔）。JP\_AnnotationStyleElement を継承する。

#### 属性

1. space:JP\_BasisForCharacterspace は、注記の文字と文字の間隔。対象縮尺で表示したときの間隔とする。型となる字隔基準は、任意に字隔を定義したい場合は、「ユーザ定義」、特定の値をとる場合は「字隔ピッチ」、字隔を指定せず、基線に対し均等割りを行いたい場合は「均等割り」の三つのうちいずれかを選択可能な共用型である。
  - ・ userDefinition: ユーザが任意に数値を定義する。数値の単位は Font 型で指定された文字列の大きさの単位とする。
  - ・ pitch: 列挙型である JP\_Interval によってあらかじめ用意された字隔から選択する。この JP\_Interval は任意に設定することが可能である。
  - ・ perCapita: 注記の基線に対して、文字を均等に配置するか否かを JP\_Flag 型によって選択する。均等割りを行う場合は on となる。なお、均等割りを選択した注記には、基線が定義されていないといけない。

### 3.2.7 JP\_Upright

注記文字列の表示を基線に対して直立に表示するか、表示範囲に対して垂直に直立に表示するか否かの別。JP\_AnnotationStyleElement を継承する。

#### 属性

1. upright:JP\_Flag は、基線に対して直立に表示する場合は on、表示範囲に対して直立に表示する場合は off とする。

### 3.2.8 JP\_Direction

注記文字列の表示を縦書きにするか横書きにするかの別。JP\_AnnotationStyleElement を継承する。

#### 属性

1. direction:JP\_Flag は、縦書きに表示する場合は on、横書きに表示する場合は off とする。

### 3.2.9 JP\_Mask

注記文字列が、他の地図記号や、他の注記文字列と重なっている場合に、マスク表示を行うかどうかの別。JP\_AnnotationStyleElement を継承する。

#### 属性

1. mask:JP\_Flag は、注記文字列が他の地図記号又は他の注記文字列と重なった際に、マスク表示を行うか否かの別である。マスク表示を行う場合は on、行わない場合は off とする。初期値は on とする。

### 3.2.10 JP\_Rotate

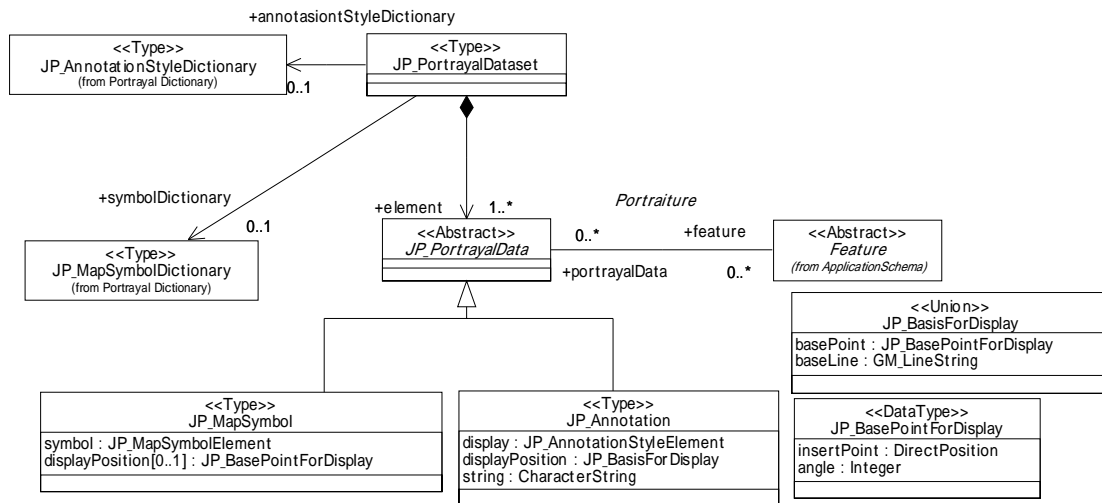
注記文字列を回転した際に表示も回転させるかどうかの別。JP\_AnnotationStyleElement を継承する。

#### 属性

1. rotate:JP\_Flag は、注記を、表示範囲の回転によらず、一定の角度を保ってもよいかどうかの別。on の場合は、角度を一定に保つことが可能（絶対座標に対して可変）、off の場合は、表示範囲の回転角度に応じて、注記の角度を変更しなくてはならない（絶対座標に対して固定）。初期値は、on とする。

## 4. 地物描画データスキーマ

地物描画データスキーマとは、各地物に必要な描画のデータの構造であり、どの地物の描画情報であるかを示す「地物への参照」と、どの辞書を使って描画するかという「描画辞書スキーマへの参照」をもつ。これにより、地物の情報と、描画のための情報を切り分けることが可能となるため、データを交換する場合は、地物データのみを交換することや、描画データを含めて交換することが可能となる。また、異なるデータ集合が同じ辞書を参照することが可能となるため、異なるデータ間で同じ地図表現を行うことができる。



図附 10-4 地物描画データスキーマ UML クラス図

### 4.1 JP\_PortrayalDataset

描画に関する情報のうち、地物インスタンスごとに必要なデータ（地物描画インスタンス）の集合。

#### 関連

1. 役割名 annotationStyleDictionary によって、この描画データ集合が参照する最大一つの注記スタイル辞書と関連付く。
2. 役割名 symbolDictionary によって、この描画データ集合が参照する最大一つの地図記号辞書と関連付く。

### 4.2 JP\_PortrayalData

地物を描画するために必要となる情報のうち地物インスタンス毎にもつべきもの。

#### 関連

1. Portraiture 関連によって、この描画インスタンスが参照する元となる地物インスタンスと関連付く。描画インスタンスは、零個以上の地物インスタンスを参照ことができ、地物インスタンスは零個以上の描画インスタンスと関連づくことができる。

### 4.3 JP\_MapSymbol

地図記号の表示に必要な情報のうち、地物インスタンス毎にもつべきもの。

#### 属性

1. `symbol` : `MapSymbolElement` によって、表示したい地図記号を地図記号辞書に定義された地図記号要素から選択し、参照する。
2. `displayPoint[0..1]` : `JP_BasePointForDisplay` によって地図記号要素を表示するための位置を定める。`JP_BasePointForDisplay` によって、挿入する座標値と角度を指定する。点記号のうち地物の空間属性をそのまま表示位置に用いる場合や、線記号、面記号などには記載しなくてもよい。

#### 関連

1. [上位型から継承する関連] `Portraiture` 関連によって、この描画インスタンスが参照する元となる地物インスタンスと関連付く。描画インスタンスは、零個以上の地物インスタンスを参照することができ、地物インスタンスは零個以上の描画インスタンスと関連づけることができる。

#### 4.4 JP\_Annotation

注記の表示に必要な情報のうち、地物インスタンス毎にもつべきもの。

#### 属性

1. `display` : `JP_AnnotationElement` は、表示したい注記のスタイル及び表示位置であり、`JP_AnnotationStyleDictionary` に定義された `JP_AnnotationElement` を選択し、参照する。
2. `displayPosition` : `JP_BasisForDisplay` は、この注記文字列を配置したい位置を定義する。型である `JP_BasisForDisplay` は、挿入点及び角度を指定する「`basePoint`」もしくは、配置する基線を `GM_LineString` によって指定する「`baseLine`」のいずれかの属性を選択する共用型である。
3. `string` : `CharacterString` は、注記として表示したい文字列である。応用スキーマに定義した他の属性値を用いたい場合は、この属性は用いずに任意に定義してよい。

#### 関連

1. [上位型から継承する関連] `Portraiture` 関連によって、この描画インスタンスが参照する元となる地物インスタンスと関連付く。描画インスタンスは、零個以上の地物インスタンスを参照することができ、地物インスタンスは零個以上の描画インスタンスと関連づけることができる。

#### 4.5 JP\_BasePointForDisplay

地図記号や注記を配置するための位置を記述する型。

#### 属性

1. `insertPoint` : `DirectPosition` は、地図記号や注記を表示したい位置の座標値を記述する。
2. `angle` : `Integer` は、地図記号や注記を回転させて配置したい場合にその指定角度を記述する。注記または地図記号を回転させる角度。上方向を Y 軸、右方向を X 軸とし、X 軸からの左回りの角度をいう（単位：度）。

#### 4.6 JP\_BasisForDisplay

注記を配置するための位置または基本となる線分を定義する共用型。

#### 属性

1. `basePoint` : `JP_BasePointForDisplay` は、挿入点と文字列の角度を指定する場合に用いる。
2. `baseLine` : `GM_LineString` は、注記の配置する線分を指定する場合に、注記の文字列を並べる線分を記述する。

## 5. 描画法 XML Schema 文書

描画法の XML Schema 文書は次の URL にあるファイル `jpsDictionary.xsd` 及び `jpsFeaturePortrayal.xsd` を使用する。

[http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0\\_2008-04](http://www.gsi.go.jp/GIS/jpgis/standardSchemas2.0_2008-04)

## 附属書 11 (規定) 空間データ製品仕様書

**序文** この附属書は、ISO 19131 (Geographic information - Data product specifications) にしたがって、このプロファイルに基づく空間データ製品仕様書の作成について規定する。

### 1. 空間データ製品仕様書

空間データ製品仕様書は、追加情報を伴う、データ集合又はデータ集合シリーズの詳細な記述であり、その作成、供給、及び別の団体による使用を可能にするものである。データ製品仕様書は論議領域の記述及び論議領域からデータ集合領域への写像のための仕様を示す。それは生産、販売、使用又はその他の目的で使われる。

### 2. 空間データ製品仕様書の記載事項

空間データ製品仕様書には、次に示す項目を記述する。

1. 概覧
2. 適用範囲
3. データ製品識別
4. データ内容及び構造
5. 参照系
6. データ品質
7. データ製品配布
8. メタデータ

また、必要に応じて次の項目を記述することができる。

9. その他

### 3. 地理情報標準プロファイルと空間データ製品仕様書記載事項との関係

このプロファイルの各項目と空間データ製品仕様書の記載事項との関係を表附 11- 1に示す。

表附 11- 1 地理情報標準プロファイルと空間データ製品仕様書記載事項の関係

空間データ製品仕様書	地理情報標準プロファイル
概覧	
適用範囲	
データ製品識別	
データ内容及び構造	6 応用スキーマのための規則
	7 空間スキーマ
	8 時間スキーマ
	9 被覆の幾何及び関数のためのスキーマ
	10 地理識別子による空間参照
	11 地物カタログ化法
参照系	附属書 2 (規定) 参照系
データ品質	附属書 3 (規定) 品質
データ製品配布	12 符号化
	附属書 8 (参考) XML に基づく符号化規則 附属書 12 (規定) 地理マーク付け言語 (GML)
メタデータ	附属書 4 (規定) メタデータ
その他	附属書 10 (参考) 描画法

### 4. 地理情報標準プロファイルに基づく空間データ製品仕様書の作成

#### 4.1 一般

このプロファイルに基づいて空間データ製品仕様書を作成する際は、空間データ製品仕様書に、このプロファイルに準拠することを明示する。

#### 4.2 プロファイルの拡張

このプロファイル拡張して使用した場合は、空間データ製品仕様書のデータ内容及び構造に、このプロファイルを拡張して使用していることを明示する。

#### 4.3 プロファイルの制限

このプロファイルの一部を制限して使用した場合は、空間データ製品仕様書のデータ内容及び構造に、このプロファイルの一部を制限して使用していることを明示する。

### 5. 空間データ製品仕様書作成マニュアル

このプロファイルに基づいて空間データ製品仕様書を作成する際は、別に定める「空間データ製品仕様書作成マニュアル」を使用することを推奨する。

## 6. 品質の要求，評価及び報告のための規則

このプロファイルに基づいて空間データ製品仕様書を作成する際，データ品質の項目は，別に定める「品質の要求，評価及び報告のための規則」を使用して作成する。



## 附属書 12 (規定) 地理マーク付け言語(GML)

**序文** この附属書は、ISO 19136(Geographic information - Geography Markup Language(GML))を使用して、箇条 本体 5.～11.に基づく応用スキーマ及び空間データを符号化する際の規則を定める。

### 1. 一般規則

#### 規則

1. ISO 19136 を用いて空間データの応用スキーマを符号化する場合には、本体に規定する要素(本体規定要素)に対応する ISO 19136 の要素を使用する。対応する要素を、表附 12-1に示す。
2. 空間データを符号化する際に使用する標準スキーマを、箇条 本体 6.～10.に規定する拡張規則に基づき拡張し、これを符号化する場合、ISO 19136 の範囲内で符号化することができる。
3. 符号化した空間データは、規則 1 及び 2 に従い作成した GML 応用スキーマに対して妥当でなければならない。

表附 12-1 JPGIS 要素に対応する ISO19136 要素

項目	本体規定要素	ISO 19136 要素
基本的なデータ型	CharacterString	xsd:string
	Boolean	xsd:boolean
	Real, Number	xsd:double
	Decimal	xsd:decimal
	Date	xsd:date
	Time	xsd:time
	DateTime	xsd:dateTime
	Integer	xsd:integer, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:positiveInteger
	Vector	gml:VectorType
	Length, Distance	gml:LengthType
	Angle	gml:AngleType
	Measure	gml:MeasureType
	Sign	gml:SignType
	UnitOfMeasure	gml:UnitOfMeasureType
空間スキーマ	GM_Object	gml:AbstractGeometry
	GM_Primitive	gml:AbstractGeometricPrimitive
	DirectPosition	gml:DirectPositionType
	GM_Position	gml:geometricPositionGroup (group)
	GM_PointArray	gml:geometricPositionListGroup (group)
	GM_Point	gml:Point
	GM_Curve	gml:Curve
	GM_Surface	gml:Surface
	GM_PolyhedralSurface	gml:PolyhedralSurface
	GM_TriangulatedSurface	gml:TriangulatedSurface
	GM_Tin	gml:Tin
GM_OrientableCurve	gml:OrientableCurve	

項目	本体規定要素	ISO 19136 要素
	GM_OrientableSurface	gml:OrientableSurface
	GM_Ring	gml:Ring
	GM_CompositeCurve	gml:CompositeCurve
	GM_Complex	gml:GeometricComplex
	GM_Aggregate	gml:MultiGeometry
	GM_CurveSegment	gml:AbstractCurveSegment
	GM_Arc	gml:Arc
	GM_ArcString	gml:ArcString
	GM_Circle	gml:Circle
	GM_LineString	gml:LineStringSegment
	GM_SurfacePatch	gml:AbstractSurfacePatch
	GM_Polygon	gml:PolygonPatch
	GM_Triangle	gml:Triangle
	TP_Object	gml:AbstractTopology
	TP_Primitive	gml:AbstractTopoPrimitive
	TP_Node	gml:Node
	TP_Edge	gml:Edge
	TP_Face	gml:Face
	TP_DirectedNode	gml:DirectedNodePropertyType
	TP_DirectedEdge	gml:DirectedEdgePropertyType
	TP_DirectedFace	gml:DirectedFacePropertyType
	TP_Complex	gml:TopoComplex
時間スキーマ	TM_Object	gml:AbstractTimeObject
	TM_Primitive	gml:AbstractTimePrimitive
	TM_GeometricPrimitive	gml:AbstractTimeGeometricPrimitive
	TM_Instant	gml:TimeInstant
	TM_Period	gml:TimePeriod
	TM_Position	gml:TimePositionType
	TM_IndeterminateValue	@TimeIndeterminateValue (attribute on TimePositionType)
	TM_TopologicalPrimitive	gml:AbstractTimeTopologyPrimitive
	TM_Node	gml:TimeNode
	TM_Edge	gml:TimeEdge
	TM_CalDate	gml:CalDate
	TM_DateAndTime	xsd:dateTime
被覆の幾何及び関数のためのスキーマ	CV_Coverage	gml:AbstractCoverage
	CV_DiscreteCoverage	gml:AbstractDiscreteCoverage
	CV_ContinuousCoverage	gml:AbstractContinuousCoverage
	CV_DiscretePointCoverage	gml:MultiPointCoverage
	CV_DiscreteGridPointCoverage	gml:GridCoverage
	CV_DiscreteCurveCoverage	gml:MultiCurveCoverage
	CV_DiscreteSurfaceCoverage	gml:MultiSurfaceCoverage
	CV_GeometryValuePair	gml:domainSet, gml:rangeSet
	CV_ValueObject	gml:domainSet, gml:rangeSet
	CV_PointValuePair	gml:domainSet, gml:rangeSet
	CV_CurveValuePair	gml:domainSet, gml:rangeSet
	CV_SurfaceValuePair	gml:domainSet, gml:rangeSet
	CV_Grid	gml:Grid

項目	本体規定要素	ISO 19136 要素
	CV_GridEnvelope	gml:GridEnvelope
	CV_GridValuesMatrix	gml:GridFunction
	CV_SequenceRule	gml:SequenceRule
	CV_SequenceType	gml:SequenceRuleNames
	CV_ThiessenPolygonCoverage	-
	CV_ThiessenValuePolygon	-
	CV_TINCoverage	-
	CV_ValueTriangle	-
地理識別子による空間参照	SI_LocationInstance	gml:LocationName
	SI_LocationType	-
	SI_Gazetteer	-
	RS_LRS	-
座標による空間参照	SC_CRS	gml:AbstractCRS
	RS_CRS	@srsName (XML attribute)
地物カタログ化法	クラス全般	-
他の規格から引用するクラスの定義	クラス全般	符号化する場合は、ISO19136 の規定に基づくこと。
四辺形グリッド被覆 (参考)	クラス全般	- (JPGIS で独自に定めた規則のため)
描画法 (参考)	クラス全般	- (JPGIS で独自に定めた規則のため)

## 2. GML 標準 XML Schema 文書

GML 標準 XML Schema の名前空間は “<http://www.opengis.net/gml/3.2>” , 名前空間接頭辞は “gml” , バージョンは “3.2.1” とする。

GML 標準 XML Schema 文書は、次のいずれかの URL にあるファイルを使用する。

[http://www.iso.org/ittf/ISO\\_19136\\_Schemas](http://www.iso.org/ittf/ISO_19136_Schemas)

<http://schemas.opengis.net/gml/3.2.1/>

**(参考資料)****ISO 19136 Geographic Information - Geography Markup Language (GML)  
附属書 E(規定)の概要****UML から GML への応用スキーマ符号化規則****E.1 一般概念**

ISO 19109 に適合する UML 応用スキーマから対応する GML 応用スキーマへの写像は、一連の符号化規則に基づく。これらの符号化規則は GML 応用スキーマの規則に適合しており、ISO 19118 に基づく。

これらの規則は ISO 19136 の 7~21, 特に ISO 19136 の 7 (GML スキーマ - 一般規則及び基本スキーマ要素) に記述されている GML モデル及び構文のための規則から導出されている。ISO 19118:2005 附属書 A の符号化規則は適用可能な場合にはいつでも使用される。

この附属書に示す規則は ISO 19109 及び ISO/TS 19103 に適合する UML 応用スキーマから (ISO 19136 の 21 (GML 応用スキーマのための規則) で定義されている規則と合致する) GML 応用スキーマへの自動的な写像を目的とする。この自動化の結果として、得られた GML 応用スキーマは XML 及び XML スキーマの能力を十分に活かしたものにはならないだろうが、明確で内容を把握しやすい XML 文法をもつ国際規格の ISO 19100 シリーズに適合した XML 実装を提供するだろう。

これらの規則は、すべての GML 応用スキーマがこれらの規則を使用して生成されなければならないということを定めていない。すべてのスキーマは ISO 19136 の 21 (GML 応用スキーマのための規則) で定義された規則に従う限り妥当及び適合した GML 応用スキーマとなり、これらは手作業、UML 応用スキーマからの自動的な導出、又は他の手法によって作成される。

スキーマ符号化規則は、XML スキーマの型及び要素宣言へ写像された応用スキーマにおけるクラス定義の一般的な考え方に基づく。これはインスタンスモデルにおけるオブジェクトが XML 文書の要素構造と合致した写像ができるようにするためである。

**E.2 符号化規則****E.2.1 一般符号化要件****E.2.1.1 応用スキーマ****E.2.1.1.1 一般(応用スキーマ, パッケージ)**

UML 応用スキーマは写像への妥当な入力のために次のすべての規則に適合しなければならない。追加要件に関しては ISO 19118:2005A.2.1 を参照。

UML 応用スキーマは ISO 19109 及び ISO/TS 19103 が定義する規則に適合しなければならない。

UML 応用スキーマはステレオタイプ<<Application Schema>>をもつパッケージによって表現されなければならない。このパッケージは GML 応用スキーマ内のオブジェクト型に写像されるすべての UML モデル要素を (すなわち、自身を直接又は間接に) 含まなければならない。パッケージは応用スキーマ内の異なる UML モデル要素をグループ化するためステレオタイプ<<Application Schema>>をもたない他のパッケージを含んでもよい。

UML モデルは完結していなければならない。次に記す例外を除き外部参照を含んではならない。事前定義クラスは国際規格の ISO19100 シリーズの規格化されたスキーマからインポートしてもよい。GML スキーマによって実装及び UML 応用スキーマによって使用される国際規格の ISO19100 シリーズのクラスは” ISO19100” の名称をもつパッケージ又はその名称をもつパッケージのいずれかの下位パッケージ内に規定しなければならない。

パッケージ間の依存関係は明確にモデル化しなければならない。ステレオタイプ<<import>>をもつ許可要素又はパッケージ間の未指定の依存要素は他のパッケージ内の要素からひとつのパッケージ内の要素の依存関係を表現するために使用しなければならない。すべての他の依存要素は無視しなければならない。図 E.1 参照。

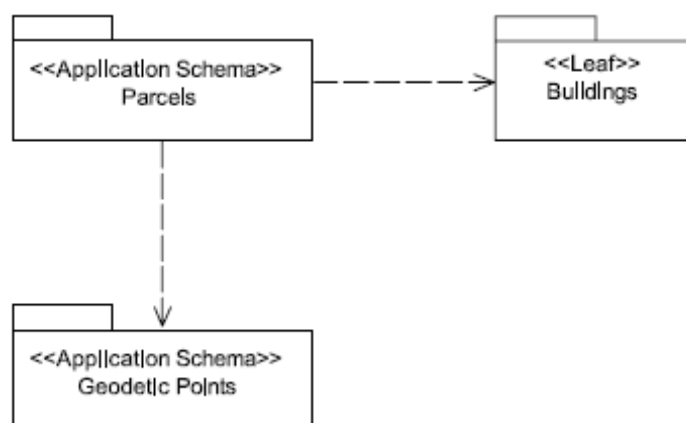


図 E.1 パッケージ間の依存関係 <参考>

すべての UML 要素の可視性は” public” に設定されなければならない。可視可能な要素のみが、応用システム間のデータ交換に使用する応用スキーマの一部でなければならない。

UML モデルにおける要素の文書化はタグ付き値” documentation” に格納されなければならない。

一意な XML 名前空間は UML 応用スキーマと関連付けられなければならない。対象とする名前空間 URI に関するタグ付き値” targetNamespace” 及び略語に関する” xmlns” は、パッケージが UML 応用スキーマを表現しているときに限り設定しなければならない。

UML 応用スキーマを表現するパッケージのバージョン番号は、適用可能な場合にはタグ付き値” version” に指定しなければならない。

GML プロファイルはタグ付き値” gmlProfileSchema” によって応用スキーマに関連付けられてもよい。設定する場合には、その値は GML プロファイルのスキーマの場所を参照する URL でなければならない。

パッケージをそれぞれの XML スキーマ文書に写像する場合、タグ付き値” xsdDocument” はスキーマ文書の妥当な相対ファイル名を示すように設定しなければならない。タグ付き値は UML 応用スキーマを表現するすべてのパッケージに関して設定しなければならない。ひとつの UML モデル内のすべてのタグ付き値” xsdDocument” は一意でなければならない。

例 タグ付き値” xsdDocument” の値は” GeodeticPoints.xsd” 又は” schema/Parcels.xsd” であってもよい。

### E.2.1.1.2 クラス

同じ応用スキーマ内のすべてのクラス名は、W3C XML Namespaces:1999 が規定するとおり、一意及び一つの“NCName”でなければならない。

地物型はステレオタイプ<<FeatureType>>をもつ UML クラスとしてモデル化しなければならない。図 E.2 参照。

備考 1 ISO 19109 及び ISO 19118:2005 附属書 A のどちらも地物型か又はオブジェクト型かの区別をしない。ISO 19109 は地物型のみを考慮するが、ISO 19118:2005 附属書 A はすべての地物型をオブジェクト型として分類する。しかしながら、この区別は GML において意味があり、また、実際に応用スキーマでしばしば必要となる。この附属書による区別は、ISO 19118:2005 附属書 A に適合する改良である。

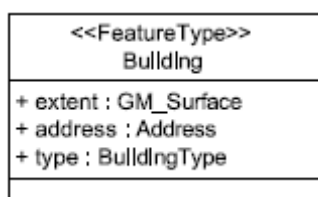


図 E.2 地物型 <参考>

オブジェクト型はステレオタイプをもたない UML クラスとしてモデル化しなければならない。オブジェクト型は、そのインスタンスが識別をもたなければならない型であるが、地物型ではない\*。

\* オブジェクト型は ISO 19109:2005 で明確には考慮されていない。それらはプロパティ型の値型としてのみ現れる。

例 オブジェクト型の例は、幾何、位相、参照系がある。これらの型のインスタンスは、例えば、名前及び識別子をもってもよい。

ステレオタイプ<<Type>>をもつ UML クラスはゼロ個以上の操作（これらは GML 応用スキーマに写像しない）、属性又は関連をもってもよい。

ステレオタイプ<<Abstract>>の使用は正しい UML 表記法の使用法と矛盾することがあるので誤解を招きやすいため、応用スキーマ内で使用すべきではない。

GML の抽象型のすべてのインスタンス化可能な下位型は地物型、オブジェクト型、またはデータ型のいずれかでなければならない。

列挙はステレオタイプ<<Enumeration>>をもつ UML クラスとしてモデル化しなければならない。

コードリストはステレオタイプ<<CodeList>>をもつ UML クラスとしてモデル化しなければならない。図 E.3 参照。

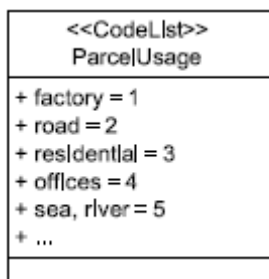


図 E.3 コードリスト &lt;参考&gt;

共用体型はステレオタイプ<<Union>>をもつ UML クラスとしてモデル化されなければならない (ISO 19107 に規定するとおり)。

他のデータ型はすべてステレオタイプ<<DataType>>をもつ UML クラスとしてモデル化しなければならない。図 E.4 参照。

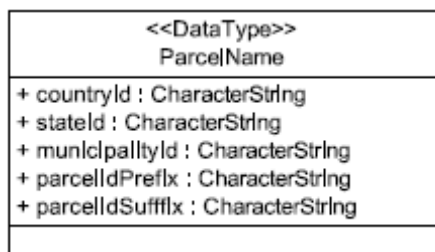


図 E.4 データ型 &lt;参考&gt;

GML プロファイルの一部となる国際規格 ISO 19100 シリーズの UML クラス及び表 D.2 の” GML 型” 列に示す GML 基本型に関する UML クラスは UML 応用スキーマで下位クラス化してもよい。下位クラスでは、追加のプロパティを加えてもよいし、下位型のプロパティに制限された多重度又は値の定義域を再定義してもよい。

備考 2 たとえプロパティの再定義が対応されても、これらの再定義されたプロパティは変換規則の中では無視されるし、再定義によって導入される制限条件の妥当性を評価するのはアプリケーションの責任である。上記に示されたものとは異なる他のステレオタイプをもつすべてのクラスは UML 応用スキーマの一部であってもよいが、無視されるだろう。

備考 3 GML スキーマによって実装された ISO 19100 シリーズの他の規則によって定義された型を応用スキーマが参照する場合、クラス名は表 D.2 の最初の列に示されたものの一つと一致することが望ましい。

表 D.2 国際規格 ISO19100 シリーズの型の実装 (ISO 19136 附属書 D より引用)

UML class	GML object element	GML type	GML property type
GM_Object	gml:AbstractGeometry	gml:AbstractGeometryType	gml:GeometryPropertyType
GM_Primitive	gml:AbstractGeometricPrimitive	gml:AbstractGeometricPrimitiveType	gml:GeometricPrimitivePropertyType
DirectPosition	-	-	gml:DirectPositionType

UML class	GML object element	GML type	GML property type
GM_Position	-	-	gml:geometricPositionGroup (group)
GM_PointArray	-	-	gml:geometricPositionListGroup (group)
GM_Point	gml:Point	gml:PointType	gml:PointPropertyType
GM_Curve	gml:Curve	gml:CurveType	gml:CurvePropertyType
GM_Surface	gml:Surface	gml:SurfaceType	gml:SurfacePropertyType
GM_PolyhedralSurface	gml:PolyhedralSurface	gml:PolyhedralSurfaceType	anonymous property type <sup>a</sup>
GM_TriangulatedSurface	gml:TriangulatedSurface	gml:TriangulatedSurfaceType	anonymous property type
GM_Tin	gml:Tin	gml:TinType	anonymous property type
GM_Solid	gml:Solid	gml:SolidType	gml:SolidPropertyType
GM_OrientableCurve	gml:OrientableCurve	gml:OrientableCurveType	gml:CurvePropertyType
GM_OrientableSurface	gml:OrientableSurface	gml:OrientableSurfaceType	gml:SurfacePropertyType
GM_Ring	gml:Ring	gml:RingType	-
GM_Shell	gml:Shell	gml:ShellType	-
-	gml:LineString	gml:LineStringType	-
-	gml:Polygon	gml:PolygonType	-
-	gml:LinearRing	gml:LinearRingType	-
GM_CompositePoint	gml:Point	gml:PointType	gml:PointPropertyType
GM_CompositeCurve	gml:CompositeCurve	gml:CompositeCurveType	anonymous property type
GM_CompositeSurface	gml:CompositeSurface	gml:CompositeSurfaceType	anonymous property type
GM_CompositeSolid	gml:CompositeSolid	gml:CompositeSolidType	anonymous property type
GM_Complex	gml:GeometricComplex	gml:GeometricComplexType	gml:GeometricComplexPropertyType
GM_Aggregate	gml:MultiGeometry	gml:MultiGeometryType	gml:MultiGeometryPropertyType
GM_MultiPoint	gml:MultiPoint	gml:MultiPointType	gml:MultiPointPropertyType
GM_MultiCurve	gml:MultiCurve	gml:MultiCurveType	gml:MultiCurvePropertyType
GM_MultiSurface	gml:MultiSurface	gml:MultiSurfaceType	gml:MultiSurfacePropertyType
GM_MultiSolid	gml:MultiSolid	gml:MultiSolidType	gml:MultiSolidPropertyType
GM_MultiPrimitive	gml:MultiGeometry	gml:MultiGeometryType	gml:MultiGeometryPropertyType
GM_CurveSegment	gml:AbstractCurveSegment	gml:AbstractCurveSegmentType	-
GM_Arc	gml:Arc	gml:ArcType	-
GM_ArcByBulge	gml:ArcByBulge	gml:ArcByBulgeType	-
-	gml:ArcByCenterPoint	gml:ArcByCenterPointType	-
GM_ArcString	gml:ArcString	gml:ArcStringType	-
GM_ArcStringByBulge	gml:ArcStringByBulge	gml:ArcStringByBulgeType	-
GM_Bezier	gml:Bezier	gml:BezierType	-
GM_BsplineCurve	gml:BSpline	gml:BSplineType	-
GM_Circle	gml:Circle	gml:CircleType	-
-	gml:CircleByCenterPoint	gml:CircleByCenterPointType	-



UML class	GML object element	GML type	GML property type
GM_Clothoid	gml:Clothoid	gml:ClothoidType	-
GM_CubicSpline	gml:CubicSpline	gml:CubicSplineType	-
GM_GeodesicString	gml:GeodesicString	gml:GeodesicStringType	-
GM_LineString	gml:LineStringSegment	gml:LineStringSegmentType	-
GM_OffsetCurve	gml:OffsetCurve	gml:OffsetCurveType	-
GM_SurfacePatch	gml:AbstractSurfacePatch	gml:AbstractSurfacePatchType	-
GM_GriddedSurface	gml:AbstractGriddedSurface	gml:AbstractGriddedSurfaceType	-
GM_ParametricCurveSurface	gml:AbstractParametricCurveSurface	gml:AbstractParametricCurveSurfaceType	-
GM_Cone	gml:Cone	gml:ConeType	-
GM_Cylinder	gml:Cylinder	gml:CylinderType	-
GM_Geodesic	gml:Geodesic	gml:GeodesicType	-
GM_Polygon	gml:PolygonPatch	gml:PolygonPatchType	-
-	gml:Rectangle	gml:RectangleType	-
GM_Sphere	gml:Sphere	gml:SphereType	-
GM_Triangle	gml:Triangle	gml:TriangleType	-
TP_Object	gml:AbstractTopology	gml:AbstractTopologyType	anonymous property type
TP_Node	gml:Node	gml:NodeType	gml:DirectedNodePropertyType
TP_Edge	gml:Edge	gml:EdgeType	gml:DirectedEdgePropertyType
TP_Face	gml:Face	gml:FaceType	gml:DirectedFacePropertyType
TP_Solid	gml:TopoSolid	gml:TopoSolidType	gml:DirectedTopoSolidPropertyType
TP_DirectedNode	-	-	gml:DirectedNodePropertyType
TP_DirectedEdge	-	-	gml:DirectedEdgePropertyType
TP_DirectedFace	-	-	gml:DirectedFacePropertyType
TP_DirectedSolid	-	-	gml:DirectedTopoSolidPropertyType
TP_Complex	gml:TopoComplex	gml:TopoComplexType	gml:TopoComplexPropertyType
-	gml:TopoPoint	gml:TopoPointType	gml:TopoPointPropertyType
-	gml:TopoCurve	gml:TopoCurveType	gml:TopoCurvePropertyType
-	gml:TopoSurface	gml:TopoSurfaceType	gml:TopoSurfacePropertyType
-	gml:TopoVolume	gml:TopoVolumeType	gml:TopoVolumePropertyType
TM_Object	gml:AbstractTimeObject	gml:AbstractTimeObjectType	anonymous property type
TM_Complex	gml:AbstractTimeComplex	gml:AbstractTimeComplexType	anonymous property type
TM_GeometricPrimitive	gml:AbstractTimeGeometricPrimitive	gml:AbstractTimeGeometricPrimitiveType	gml:TimeGeometricPrimitivePropertyType
TM_Instant	gml:TimeInstant	gml:TimeInstantType	gml:TimeInstantPropertyType

UML class	GML object element	GML type	GML property type
TM_Period	<code>gml:TimePeriod</code>	<code>gml:TimePeriodType</code>	<code>gml:TimePeriodPropertyType</code>
TM_TopologicalComplex	<code>gml:TimeTopologyComplex</code>	<code>gml:TimeTopologyComplexType</code>	<code>gml:TimeTopologyComplexPropertyType</code>
TM_TopologicalPrimitive	<code>gml:AbstractTimeTopologyPrimitive</code>	<code>gml:AbstractTimeTopologyPrimitiveType</code>	<code>gml:TimeTopologyPrimitivePropertyType</code>
TM_Node	<code>gml:TimeNode</code>	<code>gml:TimeNodeType</code>	<code>gml:TimeNodePropertyType</code>
TM_Edge	<code>gml:TimeEdge</code>	<code>gml:TimeEdgeType</code>	<code>gml:TimeEdgePropertyType</code>
TM_PeriodDuration	-	-	<code>gml:duration</code> (property element), <code>xsd:duration</code>
TM_IntervalLength	-	-	<code>gml:timeInterval</code> (group), <code>gml:TimeIntervalLengthType</code>
TM_Duration	-	-	<code>gml:timeLength</code> (group)
TM_Position	-	-	<code>gml:TimePositionType</code>
TM_IndeterminateValue	-	-	<code>@TimeIndeterminateValue</code> (attribute on <code>TimePositionType</code> )
TM_Coordinate	-	-	<code>xsd:decimal</code>
TM_CalDate	-	-	<code>gml:CalDate</code>
TM_ClockTime	-	-	<code>xsd:time</code>
TM_DateAndTime	-	-	<code>xsd:dateTime</code>
TM_Calendar	<code>gml:TimeCalendar</code>	<code>gml:TimeCalendarType</code>	<code>gml:TimeCalendarPropertyType</code>
TM_CalendarEra	<code>gml:TimeCalendarEra</code>	<code>gml:TimeCalendarEraType</code>	<code>gml:TimeCalendarEraPropertyType</code>
TM_Clock	<code>gml:TimeClock</code>	<code>gml:TimeClockType</code>	<code>gml:TimeClockPropertyType</code>
TM_CoordinateSystem	<code>gml:TimeCoordinateSystem</code>	<code>gml:TimeCoordinateSystemType</code>	anonymous property type
TM_OrdinalReferenceSystem	<code>gml:TimeOrdinalReferenceSystem</code>	<code>gml:TimeOrdinalReferenceSystemType</code>	anonymous property type
TM_OrdinalEra	<code>gml:TimeOrdinalEra</code>	<code>gml:TimeOrdinalEraType</code>	<code>gml:TimeOrdinalEraPropertyType</code>
SC_CRS	<code>gml:AbstractCRS</code>	<code>gml:AbstractCRSType</code>	<code>gml:CRSPropertyType</code>
SI_LocationInstance	-	-	<code>gml:LocationName</code>
CV_Coverage	<code>gml:AbstractCoverage</code>	<code>gml:AbstractCoverageType</code>	anonymous property type
CV_ContinuousCoverage	<code>gml:AbstractContinuousCoverage</code>	<code>gml:AbstractContinuousCoverageType</code>	anonymous property type
CV_DiscreteCoverage	<code>gml:AbstractDiscreteCoverage</code>	<code>gml:DiscreteCoverageType</code>	anonymous property type
CV_DiscretePointCoverage	<code>gml:MultiPointCoverage</code>	<code>gml:MultiPointCoverageType</code>	anonymous property type
CV_DiscreteCurveCoverage	<code>gml:MultiCurveCoverage</code>	<code>gml:MultiCurveCoverageType</code>	anonymous property type
CV_DiscreteSurfaceCoverage	<code>gml:MultiSurfaceCoverage</code>	<code>gml:MultiSurfaceCoverageType</code>	anonymous property type
CV_DiscreteSolidCoverage	<code>gml:MultiSolidCoverage</code>	<code>gml:MultiSolidCoverageType</code>	anonymous property type
CV_DiscreteGridPointCoverage	<code>gml:GridCoverage</code>	<code>gml:GridCoverageType</code>	anonymous property type

UML class	GML object element	GML type	GML property type
CharacterString	-	-	xsd:string
Boolean	-	-	xsd:boolean
Real, Number	-	-	xsd:double
Decimal	-	-	xsd:decimal
Date	-	-	xsd:date
Time	-	-	xsd:time
DateTime	-	-	xsd:dateTime
Integer	-	-	xsd:integer, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:positiveInteger <sup>b</sup>
Vector	-	-	gml:VectorType
GenericName, LocalName or ScopeName	-	-	gml:CodeType
Length, Distance	-	-	gml:LengthType
Angle	-	-	gml:AngleType
Speed	-	-	gml:SpeedType
Scale	-	-	gml:ScaleType
Area	-	-	gml:AreaType
Volume	-	-	gml:VolumeType
Measure	-	-	gml:MeasureType
Sign	-	-	gml:SignType
UnitOfMeasure	-	-	gml:UnitOfMeasureType
<sup>a</sup> GML プロパティ型に関するパターンに従う名前なし型。行内に参照または埋め込まれたオブジェクト要素は同じ行の 4 番目の列の要素である。 <sup>b</sup> 2 番目の列の多重値は、附属書 F に記載された逆写像に対応するために与えられている。			

汎化関係は次に示す二つのクラス間のいずれかでのみ規定してもよい。

- 両方とも地物型,
- 両方ともオブジェクト型, 又は
- 両方ともデータ型。

クラス間のすべての汎化関係はステレオタイプをもってはならない。他のステレオタイプをもつすべての汎化関係は無視されるだろう。UML 汎化の弁別子プロパティは空白でなければならない。

あるクラスが他のクラスの特化である場合、このクラスはただひとつの上位クラスをもつ。(多重継承の対応はしない。)

すべてのクラスはクラスの意味を規定するステレオタイプをもたなければならない。ステレオタイプをもたないクラスはオブジェクト型として取り扱われる。図 E.5 参照。

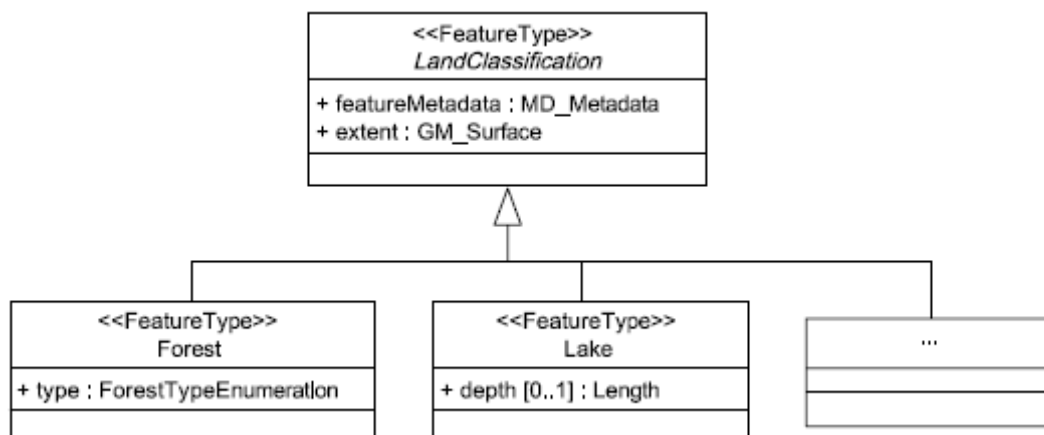


図 E.5 地物型間の汎化関係 &lt;参考&gt;

### E.2.1.1.3 属性

抽象型、地物型、オブジェクト型、データ型、または共用体型のすべての UML 属性は名前及び型をもたなければならない。名前は W3C XML Namespaces:1999 によって定義された”NCName”でなければならない。多重度は、”1”でない場合には、規定しなければならない。初期値は数字、文字列又は列挙型の属性で規定されてもよい。

型は事前定義型 (E.2.1.1.5 参照) 又は UML モデルで定義されたクラスのいずれかでなければならない。

列挙型クラスのすべての UML 属性は名前をもたなければならない。型情報は空とする。多重度、順序付け、初期値情報を属性に付与してはならない。

コードリストクラスのすべての UML 属性は名前をもたなければならない。型情報は空とする。多重度、順序付け情報は属性に付与してはならない。初期値はコードリスト値に関するコードを文書化するように規定してもよい。省略する場合、その値 (すなわち、属性の名前) をコードとして使用する。

UML クラスのプロパティは順序付けされない。UML モデルから XML スキーマへの変換時のプロパティの一貫した順序付けに対応するため、タグ付き値”sequenceNumber” (値定義域: 整数) をすべての属性に対して示さなければならない。その値はすべての属性及びクラスの関連終端に関して一意でなければならない。

### E.2.1.1.4 関連及び関連終端

すべての UML 関連は正確に二つの関連終端をもつ関連でなければならない。両方の関連終端は地物、オブジェクト、またはデータ型に接続しなければならない。また、ステレオタイプをもたないか又はステレオタイプ<<association>>をもたなければならない (そうでない場合はすべての関連は無視されるだろう。)

関連はどのようなプロパティももってはならない。

関連終端に関する規則は次の通りである。

- 関連終端が誘導可能である場合には、そのように示されなければならないし、役割名をもたなければならない。名前をもたない関連終端は、たとえ誘導可能であると示されていても無視されな

なければならない。名前がある場合は、それは W3C XML Namespaces:1999 によって定義された” NCName” でなければならない。

- 多重度は明確に与えられなければならない。
- 集成の種類は、” none” でない場合には、規定しなければならない。
- 関連終端の対象クラスがデータ型の場合、集成の種類は” 合成 “でなければならない。

図 E.6 は関連の二つの例を示している。一つの関連は双方向に誘導可能であるがもう片方は一方方向のみ誘導可能な集成である。

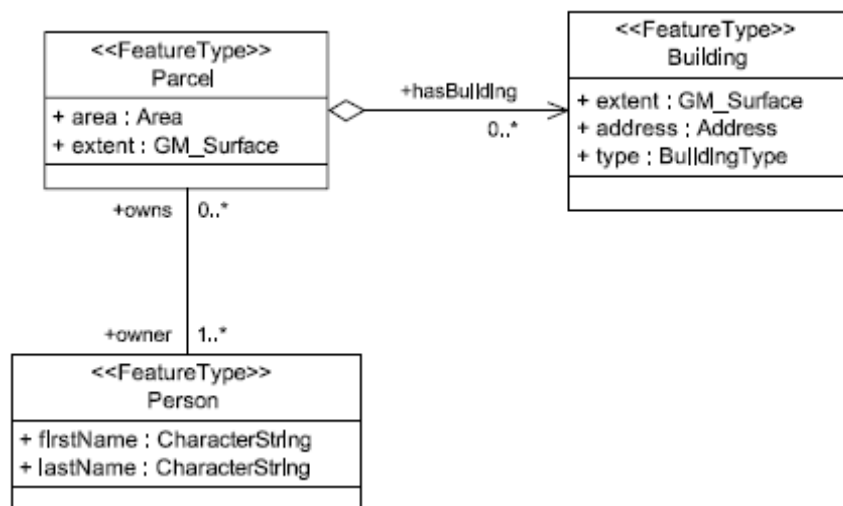


図 E.6 関連 <参考>

UML クラスのプロパティは順序付けされない。XML スキーマへの変換時に UML モデルからのプロパティの整合的な順序付けに対応するため、タグ付き値” sequenceNumber” (値定義域: 整数) はすべての関連終端に対して示されなければならない。その値はクラスのすべての属性及び関連終端に関して一意でなければならない。

#### E.2.1.1.5 事前定義型

E.2.4.4 で示された ISO/TS 19103 からの事前定義型は ISO 19118:2005 附属書 A の意味で” 基本型” として扱われる (すなわち正規の XML 符号化がそれらに付随する)。

#### E.2.1.1.6 OCL 制約

すべての OCL 制約は無視される。インスタンスモデルのこれらの制約に関する妥当性の評価は GML インスタンスを処理するアプリケーションの作業である。

備考 GML 応用スキーマを表現する XML スキーマの一部として OCL 制約を表現するために、スキマトロン言語を使用してもよい。

#### E.2.1.1.7 その他の情報

UML 応用スキーマ内の他のすべての情報は、符号化規則には使用されず、無視される。

#### E.2.1.2 文字レポトリ及び言語

XML と合致する (関連する文字レポトリをもつ) XML スキーマファイルの文字符号化には “UTF-8” 又は “UTF-16” を使用しなければならない。

### E.2.1.3 メタデータ交換

メタデータ交換は、内容モデルが E. 2. 4. 11 及び E. 2. 4. 13 に記述されている

” gml:AbstractMetadataPropertyType ” から導出される応用スキーマプロパティ要素内に規定されることにより GML インスタンス文書内のすべての地物及び地物の集まりに関して規定されてもよい。

メタデータ交換に関する特定のスキーマは GML 応用スキーマに付加されない。

### E.2.1.4 データセット及びオブジェクト識別

オブジェクトを識別するために、XML の ID 構造と合致する一意の識別子を使用する。

備考 XML の ID 構造は、それらが現れる XML 文書内でこれらの識別子が一意の識別子であることだけが必要である。

### E.2.1.5 更新の仕組み

明確な更新の仕組みは GML 応用スキーマ内で定義された地物に関しては定義されない。データ格納の更新のためには、他の仕組みの使用を想定している。

備考 一例は OpenGIS Web Feature Service Implementation Specification の ” Transaction ” 操作である。

## E.2.2 入力データ構造

入力データ構造の表記は ISO 19118:2005 A.3 を参照。

## E.2.3 出力データ構造

この符号化規則は XML Recommendation 1.0 及び XML Linking Language (XLink) Version 1.0 に基づく。交換フォーマットの構造を支配する出力データ構造に関するスキーマは XML スキーマ 1.0 及び応用スキーマ規則 (ISO 19136 の 21 (GML 応用スキーマのための規則) 参照) に合致する妥当な XML スキーマ (の集まり) でなければならない。

XML スキーマ交換規則を次の従節で定義する。

## E.2.4 交換規則

### E.2.4.1 一般概念

スキーマ交換規則は XML スキーマ文書 (XSD) が ISO 19109 に合致する UML 内で表現された応用スキーマからどのように導出されなければならないかを定義する。E2.1 に記述されたガイドラインに従う UML モデルから写像を記述するため、いくつかの一般則が E. 2. 4 に定義されている。

備考 この附属文書では、名前空間 ” xsd: ” は、XML スキーマの名前空間 ” http://www.w3c.org/2001/XMLSchema ” を参照するために使用する。名前空間 ” gml: ” は、GML の名前空間 ” http://www.opengis.net/gml/3.2 ” を参照するために使用する。

本規則は、ISO 19136 の 7~21 (特に ISO 19136 の 7 (GML スキーマ - 一般規則及び基本スキーマ要素) , 9 (GML スキーマ - 地物) , 21 (GML 応用スキーマのための規則) ) に記述された GML モデル及び構文、かつ、ISO 19118:2005 附属書 A の符号化規則に基づく。

スキーマ符号化規則は、XML スキーマの型及び要素宣言へ写像された UML 応用スキーマにおけるクラス定義の一般的な考え方に基づく。これは、インスタンスモデルにおけるオブジェクトが XML 文書の要素構造と合致した写像ができるようにするためである。

表 E.1 に概要を示す。

表 E.1 スキーマ符号化概要

表：UML→GML 応用スキーマ概要	
UML 応用スキーマ	GML 応用スキーマ
パッケージ	パッケージごとにひとつの XML スキーマ文書
<<Application Schema>>	XML スキーマ文書
<<DataType>>	その内容モデルがグローバルな範囲の XML スキーマ複合型、プロパティ型であるグローバルな要素
<<Enumeration>>	列挙値をもつ xsd:string の制限
<<CodeList>>	列挙及びパターンの共用体（既定の写像、代替写像は辞書への参照である）
<<Union>>	そのメンバーが GML オブジェクト、地物又はデータ型に対応するオブジェクトである選択グループ
<<FeatureType>>	グローバルな要素で、その内容モデルが gml:AbstractFeatureType、プロパティ型の直接又は間接の拡張によって導出されるグローバルな範囲の XML スキーマ型であるもの
ステレオタイプなし又は<<Type>>	グローバルな要素で、その内容モデルが gml:AbstractGMLType、プロパティ型の直接又は間接の拡張によって導出されるグローバルな範囲の XML スキーマ型であるもの
操作	符号化なし
属性	ローカルな xsd:element で、その型は（型が複合型の場合には）プロパティ型又は単純型のどちらかである
関連役割	ローカルな xsd:element で、その型は常にプロパティ型である（名前付けられかつ誘導可能な役割のみ）
一般的な OCL 制約	符号化なし

備考 <<Feature Type>>は、ISO/TS 19103 にも ISO 19109 にも現れない新しいステレオタイプであり、型が GF\_FeatureType の実現化及び AbstractFeature からの特化であることを示すために使用する。

属性及び関連役割の多重度は<xsd:element>宣言内の” minOccurs” 及び” maxOccurs” 属性に写像される。詳細な写像規則は後述する。

UML から XML スキーマへの写像を制御するために、異なる UML モデル要素に関しては、異なるタグ付き値を使用する。

表 E.2 タグ付き値

UML モデル要素	関連するタグ付き値
パッケージ	<ul style="list-style-type: none"> <li>— documentation</li> <li>— xsdDocument</li> <li>— targetNamespace (only &lt;&lt;Application Schema&gt;&gt;)</li> <li>— xmlns (only &lt;&lt;Application Schema&gt;&gt;)</li> <li>— version (only &lt;&lt;Application Schema&gt;&gt;)</li> <li>— gmlProfileSchema (only &lt;&lt;Application Schema&gt;&gt;)</li> </ul>
クラス	<ul style="list-style-type: none"> <li>— documentation</li> <li>— noPropertyType</li> <li>— byValuePropertyType</li> <li>— isCollection</li> <li>— asDictionary (only &lt;&lt;CodeList&gt;&gt;)</li> <li>— xmlSchemaType (only &lt;&lt;Type&gt;&gt;)</li> </ul>
属性及び関連終端	<ul style="list-style-type: none"> <li>documentation</li> <li>sequenceNumber</li> <li>inlineOrByReference</li> <li>isMetadata</li> </ul>

#### E.2.4.2 UML パッケージ

ひとつの XML スキーマ文書は、タグ付き値によって規定されるファイル名をもつタグ付き値 ”xsdDocument” を伴うパッケージごとに生成される。

タグ付き値 ”xsdDocument” がパッケージに関して設定されている場合、そのスキーマ文書はパッケージによって直接所有される UML クラスの結果であるすべての XML スキーマ構成要素を包含する。パッケージが UML 応用スキーマでない場合、そのスキーマ文書は、そのパッケージを所有するパッケージのスキーマ構成要素を含むスキーマ文書に包含されなければならない。

タグ付き値 ”xsdDocument” がパッケージに関して設定されていない場合、すべてのスキーマ構成要素は、そのパッケージを所有するパッケージのスキーマ構成要素を含むスキーマ文書内で宣言される。

備考 タグ付き値は、ステレオタイプ<<Application Schema>>をもつすべてのパッケージに対して強制的であるが、他のすべてのパッケージに対しては任意的である。

すべてのスキーマ文書に関して、ルート要素の ”targetNamespace” 及び ”version” 属性はスキーマ文書内でスキーマ構成要素を所有する UML 応用スキーマを表現するパッケージ内の別の同じ名前のタグ付き値と合致して設定されなければならない。 ”version” タグ付き値が規定されていない場合には、 ”unknown” 値を使用しなければならない。また、略語としてのタグ付き値 ”xmlns” の値をもつ対象名前空間に関して、 ”xmlns” 属性を規定しなければならない。

例1 “<http://www.myorg.com/myns>” が対象名前空間であってもよいし、 ”myns” がスキーマ文書内で使用されている関連した略語であってもよい。

ステレオタイプ <<Application Schema>> をもつパッケージのすべてのタグ付き値 ”gmlProfileSchema” に関して、タグ付き値の内容をもつ要素 <gml:gmlProfileSchema> は、ISO 19136 の 20.5 (応用スキーマから GML プロファイル参照のための規則) で規定されている <schema> 要素の appinfo annotation 内に作成されなければならない。



他のスキーマの必要なインポート及び他のスキーマ文書の付加的なインクルードを決定するためには、パッケージ間の依存性を使用しなければならない。

- 依存関係のある対象パッケージによって規定されたスキーマ構成要素が、サービス提供者パッケージのものと同じ対象名前空間にある場合には、その対象パッケージのスキーマ構成要素を規定するスキーマ文書は、” included” である。
- その他の場合、対象パッケージを含む UML 応用スキーマを表現するスキーマ文書は、” imported” である。

例 2 図 E.1 からの情報を写像すると次のような結果になってもよい。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.myorg.com/parcels"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gp="http://www.myorg.com/geodeticPoints"
xmlns:pcl="http://www.myorg.com/parcels"
xmlns:iso19115="http://www.isotc211.org/iso19115/"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="2003-07-20">
  <include schemaLocation="Buildings.xsd"/>
  <import namespace="http://www.myorg.com/geodeticPoints"
schemaLocation="GeodeticPoints.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="base/gml.xsd"/>
  <!-- ... -->
</schema>
```

#### E.2.4.3 UML クラス (一般規則)

UML クラスの承認されたステレオタイプは、ステレオタイプなし、<<Feature Type>>、<<Type>>、<<Data Type>>、<<Union>>、<<CodeList>>及び <<Enumeration>>である。すべてのクラスは対応するクラス分類に写像されるだろう。他のステレオタイプをもつすべての UML クラスは無視されるだろう。

すべての UML クラスは上位型をもたないか又はひとつの上位型をもたなければならない。

すべての UML クラスは名前付けされた型に写像される。型の名前には添え字” Type” を付加する。

#### E.2.4.4 UML クラス (基本型)

表 D.2 (” CharacterString” で始まる) の左列に示された ISO/TS 19103 の GML プロファイルからの基本型は、事前定義され、ISO 19109 に適合する応用スキーマ内の属性のデータ型として使用されてもよい。XML スキーマ(”xsd:”)又は GML (”gml:”) の作り付け型への写像を規定する。複数の名前が表の欄内にある場合は、太字体の名前を写像の既定型として使用しなければならない。

備考 右列の多重値は、附属書 F の逆写像に対応するためにも使用される。

例 ISO/TS 19103 Integer は”xsd:integer” に写像する。

ステレオタイプ<<Type>>をもつクラスが（例えば XML スキーマからの）正規の XML スキーマ符号化をもつ場合には、そのデータ型に対応する XML スキーマ型名が、タグ付き値”xmlSchemaType”の値として与えられなければならない。

備考 正規の符号化は、多くの場合、標準 UML-to-GML 符号化規則に従って構造化された符号化にするほうがよい。例えば、“simpleContent”に基づく簡潔な構造は、応用値域の中で既によく知られたものである。

#### E.2.4.5 UML クラス（データ型）

ステレオタイプ<<Data Type>>をもつ UML クラスは、XML スキーマ複合型に写像されなければならない。

備考 他のステレオタイプ（例えば<<Enumeration>>、<<CodeList>>、<<Union>>又は事前定義された基本型）をもつデータ型は、別に扱われる。E.2.4.4、E.2.4.8、E.2.4.9、E.2.4.10 を参照。

クラスが上位型をもたない場合、それは XML スキーマ内の非導出型である。そうでない場合は、その”gml:AbstractGMLType”から（直接又は間接に）導出されるべきではない上位型を拡張する。属性も誘導可能な関連役割ももたない抽象上位型は無視される。

名前（UML クラスの名前）、型（UML クラスの名前及び”型”）、（クラスが抽象型の場合には）抽象性、置換グループ（上位クラスの正規の要素名前か又は、クラスが上位型をもたない場合には、gml:AbstractObject）の適切な設定をもつグローバルな XML 要素が、これらのクラスに関して定義されなければならない。

クラスが値”true”のタグ付き値”noPropertyType”をもたない場合には、名前をもつ複合型が、これらのクラス（添え字”PropertyType”を伴うクラスの名前をもつ）に関して作成されなければならない。その型は、GML で定義された関連プロパティに関するパターンに従うが（ISO 19136 の 7.2.3（GML プロパティ）参照）、Xlink 属性は許可しない。

例 図 E.4 からのデータ型”ParcelName”は次のように記述される：

```
<complexType name="ParcelNameType">
  <sequence>
    <element name="countryId" type="string"/>
    <element name="stateId" type="string"/>
    <element name="municipalityId" type="string"/>
    <element name="parcelIdPrefix" type="string"/>
    <element name="parcelIdSuffix" type="string" minOccurs="0" />
  </sequence>
</complexType>

<element name="ParcelName" type="ex:ParcelNameType"
  substitutionGroup="gml:AbstractObject"/>
<complexType name="ParcelNamePropertyType">
  <sequence>
    <element ref="ex:ParcelName"/>
  </sequence>
</complexType>
```

```

    <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

```

#### E.2.4.6 UML クラス (地物型)

ステレオタイプ<<FeatureType>>をもつ UML クラスは直接又は間接に gml:AbstractFeatureType から導出される。クラスが上位型をもたない場合、それは直接 gml:AbstractFeatureType を拡張する。そうでない場合、それは gml:AbstractFeatureType から導出されなければならない上位型を（直接又は間接に）拡張する。

- 名前 (UML クラスの名前) , 型 (UML クラスの名前及び”型” ) , 抽象性 (クラスが抽象型の場合には真) , 置換グループ (上位クラスの名前又は gml:AbstractFeature) の適切な設定をもつグローバルな XML 要素が、これらのクラスに関して定義される。
- 対象クラスの集成または合成である一方向の関連をクラスがもつ場合、その関連役割はプロパティ要素に変換され、クラスは値” true” をもつタグ付き値” isCollection” をもち、属性グループ gml:AggregationAttributeGroup が、地物型の複合型に付加される。
- クラスが値” true” をもつタグ付き値” noPropertyType” をもたない場合、これらのクラス (添え字” PropertyType” を伴うクラスの名前をもつ) に関して、名前付き複合型が作成されなければならない。その型は、GML で定義された関連プロパティに関するパターンに従う。ISO 19136 の 7.2.3 (GML プロパティ) 参照。
- クラスが値” true” をもつタグ付き値” byValuePropertyType” をもつ場合、(添え字” PropertyByValueType” を伴うクラスの名前をもつ) これらのクラスに関して、名前付き複合型が作成されなければならない。その型は、” byvalue” フォームに制限された GML 内で定義される関連プロパティに関するパターンのプロファイルである。この点についても、ISO 19136 の 7.2.3 (GML プロパティ) 参照。

例 図 E.2 による” Building” は、次のように写像される：

```

<complexType name="BuildingType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="extent" type="gml:SurfacePropertyType"/>
        <element name="address" type="pcl:AddressPropertyType"/>
        <element name="type" type="pcl:BuildingTypeType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="BuildingPropertyType">
  <sequence minOccurs="0">
    <element ref="pcl:Building"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

```

```

<complexType name="BuildingPropertyByValueType">
  <sequence>
    <element ref="pcl:Building"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

<element name="Building" type="pcl:BuildingType" substitutionGroup="gml:AbstractFeature"/>

```

#### E.2.4.7 UML クラス (オブジェクト型)

ステレオタイプをもたないか又はステレオタイプ<<Type>>をもつ UML クラスは、直接又は間接に gml:AbstractGMLType から導出される。クラスが上位型をもたない場合は、直接に gml:AbstractGMLType を拡張する。そうでない場合は、gml:AbstractGMLType から導出されなければならない上位型を（直接又は間接に）拡張する。ただし、その上位型は、（直接又は間接にも）gml:AbstractFeatureType からは導出されるべきではない。

- 名前 (UML クラスの名前) , 型 (UML クラス及び”型”の名前) , 抽象性 (クラスが抽象型の場合は、真) , 置換グループ (上位型の名前又は AbstractGML) の適切な設定をもつグローバルな XML 要素が、これらのクラスに関して定義される。
- 対象クラスの集成または合成である一方向の関連をクラスがもつ場合、その関連役割はプロパティ要素に変換され、クラスは値”ture”をもつタグ付き値”isCollection”をもち、属性グループ gml:AggregationAttributeGroup はオブジェクト型の複合型に付加される。
- クラスが値”true”をもつタグ付き値”noPropertyType”をもたない場合、これらのクラス (添え字”PropertyType”を伴うクラスの名前をもつ) に関して名前付きの複合型が作成されなければならない。その型は GML で定義された関連プロパティに関するパターンに従う (ISO 19136 の 7.2.3 (GML プロパティ) 参照)。
- クラスが値”true”をもつタグ付き値”byValuePropertyType”をもつ場合、名前付きの複合型は (添え字”PropertyByValueType”を伴うクラスの名前をもつ) これらのクラスに関して作成されなければならない。その型は”byvalue”フォームに制限された GML 内で定義された関連プロパティに関するパターンのプロファイルである (ISO 19136 の 7.2.3 (GML プロパティ) 参照)。

例

```

<element name="Ellipse" type="ex:EllipseType" substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="EllipseType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <element name="center" type="gml:DirectPositionType"/>
        <element name="semiminor" type="gml:VectorType"/>
        <element name="semimajor" type="gml:VectorType"/>
      </sequence>
    </extension>
  </complexContent>

```

```
</complexType>
```

#### E.2.4.8 UML クラス (列挙型)

ステレオタイプ<<Enumeration>>をもつ UML クラスは XML スキーマ単純型に写像される。基本型は” string” であり、値定義域は UML クラスの属性名前によって定義された文字値の集合に制限される。

例

```
<simpleType name="SignType">
  <restriction base="string">
    <enumeration value="-"/>
    <enumeration value="+"/>
  </restriction>
</simpleType>
```

#### E.2.4.9 UML クラス (コードリスト)

ステレオタイプ<<CodeList>>をもち、かつ値” true” のタグ付き値” asDictionary” をもたない UML クラスは、列挙型のように写像されなければならないが、次の相違がある。

- ファセット<pattern value=' other: \ w {2,}' />” は事前定義された値に加えどのような文字値も許可することを付加されなければならない。これらの自由な値は” other:” を接頭辞として付ける。
- コードがコードリスト値に関して規定される場合は、ただそのコードのみが列挙ファセットとして表現されなければならない。
- 符号化されたコード値は、列挙された値の文字値を規定する gml:description 要素をもつ appinfo annotation をもち適格化されなければならない。

例 1 図 E.3 からのコードリスト” ParcelUsage” は次のように表現されてもよい：

```
<simpleType name="ParcelUsageType">
  <union memberTypes="pcl:ParcelUsageEnumerationType pcl: ParcelUsageOtherType"/>
</simpleType>

<simpleType name="ParcelUsageEnumerationType">
  <restriction base="string">
    <enumeration value="1">
      <annotation>
        <appinfo><gml:description>factory</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="2">
      <annotation>
        <appinfo><gml:description>road</gml:description></appinfo>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>
```

```

    <enumeration value="3">
      <annotation>
        <appinfo><gml:description>residential</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="4">
      <annotation>
        <appinfo><gml:description>offices</gml:description></appinfo>
      </annotation>
    </enumeration>
  <enumeration value="5">
    <annotation>
      <appinfo><gml:description>sea,
        river</gml:description></appinfo>
    </annotation>
  </enumeration>
</restriction>
</simpleType>

<simpleType name="ParcelUsageOtherType">
  <restriction base="string">
    <pattern value="other: ¥w{2,}" />
  </restriction>
</simpleType>

```

代案としては、クラスが値” true” をもつタグ付き値” asDictionary” をもつ場合には、gml:Dictionary がコードリストを表現するために使われなければならない。

例 2 図 E.3 からのコードリスト” ParcelUsage” は GML 辞書文書内で次のように表現されなければならない：

```

<gml:Dictionary gml:id="CodeList" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/gml/3.2
  gml.xsd">
  <gml:identifier codeSpace="http://www.someorg.de/cl.xml">My code
  lists</gml:identifier>
  <gml:dictionaryEntry>
    <gml:Dictionary gml:id="ParcelUsage">
      <gml:identifier
        codeSpace="http://www.someorg.de/cl.xml">ParcelUsage</gml:identifier>
      <gml:dictionaryEntry>
        <gml:Definition gml:id="ParcelUsage_1">

```

```

        <gml:description>factory</gml:description>
        <gml:identifier
          codeSpace="http://www.someorg.de/cl.xml#ParcelUs
          age">1</gml:identifier>
      </gml:Definition>
    </gml:dictionaryEntry>
    <gml:dictionaryEntry>
      <gml:Definition gml:id="ParcelUsage_2">
        <gml:description>road</gml:description>
        <gml:identifier
          codeSpace="http://www.someorg.de/cl.xml#ParcelUs
          age">2</gml:identifier>
      </gml:Definition>
    </gml:dictionaryEntry>
    <gml:dictionaryEntry>
      <gml:Definition gml:id="ParcelUsage_3">
        <gml:description>residential</gml:description>
        <gml:identifier
          codeSpace="http://www.someorg.de/cl.xml#ParcelUs
          age">3</gml:identifier>
      </gml:Definition>
    </gml:dictionaryEntry>
    <gml:dictionaryEntry>
      <gml:Definition gml:id="ParcelUsage_4">
        <gml:description>offices</gml:description>
        <gml:identifier
          codeSpace="http://www.someorg.de/cl.xml#ParcelUs
          age">4</gml:identifier>
      </gml:Definition>
    </gml:dictionaryEntry>
    <gml:dictionaryEntry>
      <gml:Definition gml:id="ParcelUsage_5">
        <gml:description>sea, river</gml:description>
        <gml:identifier
          codeSpace="http://www.someorg.de/cl.xml#ParcelUs
          age">5</gml:identifier>
      </gml:Definition>
    </gml:dictionaryEntry>
  </gml:Dictionary>
</gml:dictionaryEntry>
</gml:Dictionary>

```

インスタンス文書で参照は次の例のように符号化されるだろう（内容モデルとして gml:CodeType を使用する。E. 2. 4. 11 参照）：

```
<usage codeSpace="http://www.someorg.de/example/cl.xml#ParcelUsage">1</usage>
```

この codeSpace 属性は辞書を示し、その値は辞書内の登録の名前である。

コードリストが GML 応用スキーマ内へ符号化される方法は、その値定義域としてコードリストをもつプロパティ要素がどのように符号化されなければならないかも決定する。E.2.4.11 参照。

#### E.2.4.10 UML クラス (共用体)

ステレオタイプ<<Union>>をもつ UML クラスは XML スキーマ複合型として写像される。これらのクラスはデータ型 (E.2.4.5 参照) のように写像されるが、正確にプロパティのひとつが共用体のインスタンス内に規定されるようになるために、プロパティの<xsd:sequence>の代わりに<xsd:choice>が使用される。

例

```
<complexType name="RemoteResourceType">
  <choice>
    <element name="name" type="string"/>
    <element name="uri" type="anyURI"/>
  </choice>
</complexType>
```

#### E.2.4.11 UML 属性及び関連役割

UML 属性又はオブジェクト若しくは地物型の関連役割は、オブジェクト又は地物型の内容モデルを定義する複合型内の同じ名前をもつローカルな要素に写像される。minOccurs 及び maxOccurs 属性は、UML モデル内の定義と合致するよう設定される (写像の詳細に関しては ISO 19118:2005 附属書 A を参照)。その型は、UML 内のプロパティの値の型に依存する。

プロパティの値の型が単純型内容である場合、その型は直接使用される。

例 1 <element name="count" type="integer"/>

プロパティの値の型が複合型内容である場合、プロパティ型を使用しなければならない。プロパティ型の既定の符号化は、地物及びオブジェクト型のインライン又は参照渡し表現及びデータ及び共用体型のインライン表現の両方を許可する。地物及びオブジェクト型はそれぞれ値” inline” 又は” byReference” をもつタグ付き値” inlineOrByReference” を用いてインライン又は参照渡しに制限されてもよい。そのタグ付き値が欠落しているか又はその値が” inlineOrByReference” の場合には、既定の符号化を使用しなければならない。

属性又は関連役割がメタデータプロパティである場合、プロパティ型は gml:AbstractMetadataPropertyType を示さなければならない (ISO 19136 の 7.2.6 (メタデータ) 参照)。またメタデータプロパティは、値” true” をもつタグ付き値” isMetadata” をもつか又はその値が ISO 19115:2003 によって定義されたクラスであるプロパティである。関連役割が集成か又は合成の対象端である場合、プロパティ型は、それがメタデータプロパティでない場合は、gml:AbstractMemberType (ISO 19136 の 7.2.5.1 (AbstractMemberType 及び導出されたプロパティ型) 参照) を示さなければならない。関連役割が合成か又はオブジェクトに値付けされた属性の対象



端である場合、そのプロパティ要素は、gml:OwnershipAttributeGroup の” owns” 属性が” true” であること主張するスキマトロン制約を付加しなければならない。そのスキマトロン制約は、次のパターンに従わなければならない。

```
<sch:pattern>
  <sch:rule context="qualified name of the object element">
    <sch:report test="qualified property name/@owns='true'">This
      property is a composition, values must be owned</sch:report>
    </sch:rule>
  </sch:pattern>
```

例2 位置を記述する点オブジェクトを制御する地物型 ex:MyFeature のプロパティ ex:representativeLocation に関して、これを次のように記述することが考えられる。

```
<element name="representativeLocation" type="gml:PointPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern">
        <sch:rule context="ex:MyFeature">
          <sch:report
            test="ex:representativeLocation/@owns='true'">Th
              is property is a composition, values must be
              owned</sch:report>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
```

プロパティ型が既にその応用スキーマ内で名前付きの型として規定されている場合（これはタグ付き値” noPropertyType” 及び” byValuePropertyType” を調査することにより見つけることができる）、このスキーマ構成要素が参照されなければならない。そうでない場合は、匿名プロパティ型が、プロパティ要素内にローカルに定義されなければならない。

符号化プロパティが関連終端で、他の関連の関連終端もまた GML 応用スキーマ内で符号化される場合、他の関連終端のプロパティ名前は、プロパティ要素の appinfo annotation 内の gml:reversePropertyName 要素内に符号化されなければならない（ISO 19136 の 7.2.3.9（同じ関連で表現するプロパティ）参照）。

例3 参照渡し又はインライン

```
<element name="owner" type="ex:PersonPropertyType" minOccurs="0" >
  <annotation>
    <appinfo>
      <gml:reversePropertyName>ex:owns</gml:reversePropertyName>
    </appinfo>
```

```

    </annotation>
</element>
...
<complexType name="PersonPropertyType">
  <sequence minOccurs="0">
    <element ref="ex:Person"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

```

又は

```

<element name="owner" minOccurs="0" >
  <annotation>
    <appinfo>
      <gml:reversePropertyName>ex:owns</gml:reversePropertyName>
    </appinfo>
  </annotation>
  <complexType>
    <sequence minOccurs="0">
      <element ref="ex:Person"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</element>

```

あるいは、プロパティ型は、タグ付き値” inlineOrByReference” に依存して、インライン又は参照渡しのひとつの表現のみに対応してもよい。

例4 インラインのみ

```

<element name="owner" type="ex:PersonPropertyByValueType" minOccurs="0" />
...
<complexType name="PersonPropertyByValueType">
  <sequence>
    <element ref="ex:Person"/>
  </sequence>
</complexType>

```

又は

```

<element name="owner" minOccurs="0" >
  <complexType>
    <sequence>
      <element ref="ex:Person"/>
    </sequence>
  </complexType>

```

```

    </sequence>
  </complexType>
</element>

```

参照渡し表現のみが対応されるべきである場合、プロパティ要素は、対象型の正規の要素名前を規定する appinfo annotation 要素 gml:targetElement をもたなければならない。

```
<element name="targetElement" type="string"/>
```

符号化された要素が関連終端であり、かつ他の関連の関連終端もまた GML 応用スキーマ内で符号化される場合、他の関連終端のプロパティ名前は、前記で規定されたもう一つの appinfo annotation 要素 gml:reversePropertyName 内で符号化されなければならない。

#### 例 5 参照渡しのみ

```

<element name="owner" type="gml:ReferenceType" minOccurs="0" >
  <annotation>
    <appinfo>
      <gml:targetElement>ex:Person</gml:targetElement>
      <gml:reversePropertyName>ex:owns</gml:reversePropertyName>
    </appinfo>
  </annotation>
</element>

```

クラスの符号化に依存して、コードリスト又は列挙型の UML 属性は、string 値（値定義域：列挙又はコードリストの値）又は対応する辞書登録を参照する値のどちらかを伴う要素に写像されなければならない。インスタンス内では、辞書は codeSpace 属性を用いて明確に参照されてもよい。辞書を参照する URI に関する既定値は appinfo annotation 要素 gml:defaultCodeSpace を用いて提供されてもよい。

```
<element name="defaultCodeSpace" type="anyURI"/>
```

例 6 コードリスト "BuildingType" は次のように表現される。

```
<element name="type" type="ex:BuildingTypeType"/>
```

又は

```

<element name="type" type="gml:CodeType">
  <annotation>
    <appinfo>
      <gml:defaultCodeSpace>http://www.someorg.de/example/cl.xml#BuildingType</gml:defaultCodeSpace>
    </appinfo>
  </annotation>
</element>

```

UML 属性又は UML 関連役割が再定義される場合（すなわち、下位クラスが上位クラス内と同じ名前をもつ関連又は関連役割を含む）、このプロパティは、下位型の内容モデルの部分ではない。インスタ

ンスが概念モデル内で表現されるそのような制約と適合することを主張するのは、アプリケーションの責任である。

クラスのすべての属性及び関連役割は、タグ付き値” sequenceNumber” の昇順分類順序で変換されなければならない。

#### E.2.4.12 文書化

UML モデルの要素からのタグ付き値” documentation” は、XML スキーマファイル内の注記／文書化要素に写像されなければならない。

例

```
<element name="curveProperty" type="gml:CurvePropertyType">
  <annotation>
    <documentation>This property element either references a curve via the
      XLink-attributes or contains the curve element. curveProperty is the
      predefined property which can be used by GML application schemas whenever
      a GML feature has a property with a value that is substitutable for
      AbstractCurve.</documentation>
    </annotation>
  </element>
```

#### E.2.4.13 国際規格 ISO 19100 シリーズからインポートされたクラス

UML 応用スキーマが国際規格 ISO 19100 シリーズからクラスをインポートする場合、前記で定義された規則に加え、次の規則を適用する。

GML スキーマによって実装された国際規格 ISO 19100 シリーズからのクラスは認定されなければならない。国際規格 ISO 19100 シリーズからのクラスの使用は ISO 19109 と適合しなければならない。国際規格 ISO 19100 シリーズから関係のあるクラスの写像は表 D.2 に示されている。

ISO 19115 からクラス及び ISO/TS 19139 で実装されたクラスがプロパティの型として使われる場合、gml:AbstractMetadataType を示す無名のプロパティ型が定義されなければならない。カプセル化されたオブジェクト要素は、ISO/TS 19139 で規定されたメタデータ型に関して対応するオブジェクト要素である。

#### E.2.4.14 事前定義された XML 符号化をもつ他の概念モデルからインポートされたクラス

標準的な XML 符号化が既に規定されている他の UML モデルから、UML 応用スキーマがクラスをインポートする場合、前記で定義された規則に加え、次の規則を適用する。

表 D.2 のインポートされたクラスに関する拡張を規定しなければならない。その表は、UML 内の応用スキーマと一緒に配布されなければならない。

インポートされたモデルから XML スキーマへ関係のあるクラスの写像は、規定的にこの表によって規定される。

E.3 例 <参考>

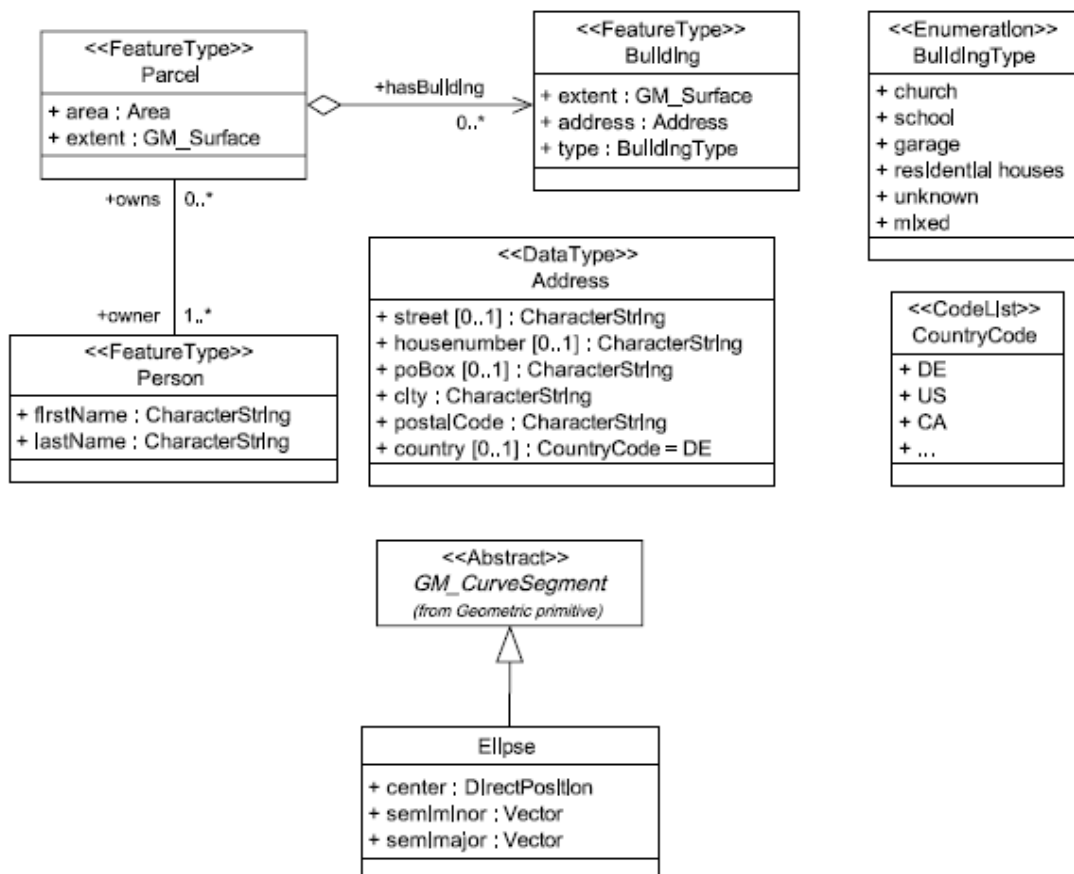


図 E.7 応用スキーマ例

図 E.7 の応用スキーマは、次のように符号化されてもよい：

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml/3.2"
elementFormDefault="qualified"
version="1.0">
  <!-- ===== -->
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="./gml.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink" schemaLocation="./xlinks.xsd"/>
  <!-- ===== -->
  <element name="Parcel" substitutionGroup="gml:AbstractFeature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="area"
              type="gml:AreaType"/>
            <element name="extent"
              type="gml:SurfacePropertyType"/>
          
```

```

                <element name="owner"
                    type="ex:PersonPropertyType"
                    maxOccurs="unbounded">
                    <annotation>
                        <appinfo><gml:reverseProperty>e
                            x:owns</gml:reverseProperty></appinfo>
                        </annotation>
                    </element>
                <element name="hasBuilding"
                    type="ex:BuildingPropertyType"
                    minOccurs="0"
                    maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
</element>
<complexType name="ParcelPropertyType">
    <sequence minOccurs="0">
        <element ref="ex:Parcel"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
<!-- ===== -->
<element name="Building" substitutionGroup="gml:AbstractFeature">
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="extent"
                        type="gml:SurfacePropertyType"/>
                    <element name="address">
                        <complexType>
                            <sequence>
                                <element
                                    name="Address"
                                    type="ex:AddressType"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="type"
                        type="ex:BuildingTypeType"/>
                </sequence>
            </extension>
        </complexContent>
    </element>

```

```

    </complexType>
</element>
<complexType name="BuildingPropertyType">
  <sequence minOccurs="0">
    <element ref="ex:Building"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
<!-- ===== -->
<element name="Person" substitutionGroup="gml:AbstractFeature">
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="firstName" type="string"/>
          <element name="lastName" type="string"/>
          <element name="owns" type="ex:ParcelPropertyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
              <appinfo><gml:reverseProperty>ex:owner</gml:reverseProperty></appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<complexType name="PersonPropertyType">
  <sequence minOccurs="0">
    <element ref="ex:Person"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
<!-- ===== -->
<complexType name="AddressType">
  <sequence>
    <element name="street" type="string" minOccurs="0"/>

```

```

        <element name="hounumber" type="string" minOccurs="0"/>
        <element name="poBox" type="string" minOccurs="0"/>
        <element name="city" type="string"/>
        <element name="postalCode" type="string"/>
        <element name="country" type="ex:CountryCodeType" minOccurs="0"
        default="DE"/>
    </sequence>
</complexType>
<!-- ===== -->
<simpleType name="BuildingTypeType">
    <restriction base="string">
        <enumeration value="church"/>
        <enumeration value="school"/>
        <enumeration value="garage"/>
        <enumeration value="residential houses"/>
        <enumeration value="unknown"/>
        <enumeration value="mixed"/>
    </restriction>
</simpleType>
<!-- ===== -->
<simpleType name="CountryCodeType">
    <union memberTypes="ex:CountryCodeEnumerationType
    ex:CountryCodeOtherType"/>
</simpleType>
<simpleType name="CountryCodeEnumerationType">
    <restriction base="string">
        <enumeration value="DE"/>
        <enumeration value="US"/>
        <enumeration value="CA"/>
        <enumeration value="..."/>
    </restriction>
</simpleType>
<simpleType name="CountryCodeOtherType">
    <restriction base="string">
        <pattern value="other: ¥w{2,}"/>
    </restriction>
</simpleType>
<!-- ===== -->
<element name="Ellipse" type="ex:EllipseType"
substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="EllipseType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>

```



(参考資料)

```

        <element                                name="center"
        type="gml:DirectPositionType"/>
        <element                                name="semiminor"
        type="gml:VectorType"/>
        <element                                name="semimajor"
        type="gml:VectorType"/>
    </sequence>
</extension>
</complexContent>
</complexType>
</schema>
```