

Проект



**СТАНДАРТ МІНІСТЕРСТВА ОХОРОНИ НАВКОЛИШНЬОГО  
ПРИРОДНОГО СЕРЕДОВИЩА УКРАЇНИ**

---

**Комплекс стандартів “Географічна інформація / геоматика”**

**ОБМІННИЙ ФОРМАТ ГЕОПРОСТОРОВИХ ДАНИХ НА ОСНОВІ  
ГЕОГРАФІЧНОЇ МОВИ РОЗМІТКИ GML  
(ISO 19136:2007)**

**СОУ ISO 19136:2009**

*Видання офіційне*

**Київ  
МІНПРИРОДИ  
200\***

## ПЕРЕДМОВА

**1. РОЗРОБЛЕНО** Науково-дослідний інститут геодезії і картографії Міністерства охорони навколишнього природного середовища України (НДІГК)

РОЗРОБНИКИ: **Ю. Карпінський**, д-р техн. наук, проф. (керівник розробки),  
**А. Лященко**, д-р техн. наук, проф., **В. Омельченко**, **Р. Осьмак**, **Р Рунець**.

**2. ВНЕСЕНО** Технічним комітетом зі стандартизації ТК 103 "Географічна інформація / геоматика"

**3. ПРИЙНЯТО ТА НАДАНО ЧИННОСТІ:** наказ Міністерства охорони навколишнього природного середовища України від \_\_\_\_ \_\_\_\_ 200\_ р. № \_\_\_\_

**4. ЗАРЕЄСТРОВАНО:** Українським науково-дослідним і навчальним центром проблем стандартизації, сертифікації та якості (УкрНДНЦ), № \_\_\_\_ від \_\_\_\_

**5. УВЕДЕНО ВПЕРШЕ**

---

**Цей стандарт не може бути повністю або частково відтворений, тиражований і розповсюджений як офіційне видання без дозволу Міністерства охорони навколишнього природного середовища України**

Мінприроди 2009

## Зміст

1 СФЕРА ЗАСТОСОВНОСТІ.....	1
2 ВІДПОВІДНІСТЬ .....	1
2.1 Вимоги до відповідності .....	1
2.2 Класи відповідності, пов'язані із GML прикладними схемами .....	2
2.3 Класи відповідності, зв'язані з GML профілями.....	2
2.4 Класи відповідності, зв'язані з GML документами.....	4
2.5 Класи відповідності, пов'язані із реалізацією у програмному забезпеченні ....	4
3 НОРМАТИВНІ ПОСИЛАННЯ .....	4
4 ТЕРМІНИ ТА ПОЗНАЧЕННЯ .....	5
4.1 ТЕРМІНИ ТА ВИЗНАЧЕННЯ .....	5
4.2 Символи та скорочення .....	11
5 ПОЗНАКИ ТА СКОРОЧЕННЯ .....	11
5.1 XML простори імен .....	11
5.2 Облік версій.....	12
5.3 Скасовані частини попередніх версій GML.....	12
5.4 UML нотації.....	12
5.5 XML схема .....	14
6 ОГЛЯД GML СХЕМИ .....	14
6.1 GML схема .....	14
6.2 GML прикладні схеми.....	15
6.3 Відношення між комплексом міжнародних стандартів серії ISO 19100, GML схемою та GML прикладними схемами.....	15
6.4 Перетворення компонентів GML схеми в документи схеми.....	16
6.5 Скасовані та експериментальні компоненти схеми .....	18
7 GML СХЕМА – ЗАГАЛЬНІ ПРАВИЛА ТА ОСНОВНІ КОМПОНЕНТИ СХЕМИ.....	18
7.1 МОДЕЛЬ ТА СИНТАКСИС GML .....	18
7.1.1 Документи GML екземплярів .....	18
7.1.2 Лексичні домовленості.....	19
7.1.3 Визначення мови GML у XML схемі.....	20
7.2 КОМПОНЕНТИ СХЕМИ GMLBASE .....	20
7.2.1 Цілі компонентів базової схеми .....	20
7.2.2 Базові об'єкти .....	20
7.2.3 GML властивості.....	22
7.2.4 Стандартні властивості GML об'єктів.....	27
7.2.5 GML колекції об'єктів.....	28
7.2.6 Метадані .....	30
8 GML СХЕМА – XLINK ПОСИЛАННЯ ТА БАЗОВІ ТИПИ.....	33
8.1 Посилання – асоціації об'єктів та віддалені властивості .....	33
8.2 Базові типи.....	35
8.2.1 Огляд.....	35
8.2.2 Зв'язок з ISO/TS 19103.....	36
8.2.3 Прості типи .....	36
9 GML СХЕМА – ГЕОГРАФІЧНІ ОБ'ЄКТИ.....	45

9.1	Загальні концепції.....	45
9.2	Зв'язок з ISO 19109.....	45
9.3	Об'єкти.....	45
9.4	Стандартні властивості об'єкту.....	46
9.5	Властивості геометрії.....	48
9.6	Властивості топології.....	49
9.7	Часові властивості.....	50
9.8	Визначення специфічно-прикладних типів географічних об'єктів.....	51
9.9	Колекції об'єктів.....	52
9.9.1	GML колекції географічних об'єктів.....	52
9.9.2	AbstractFeatureMemberType та похідні типи властивості.....	54
9.10	ПРОСТОРОВА РЕФЕРЕНЦНА СИСТЕМА, ВИКОРИСТАНА В ГЕОГРАФІЧНИХ ОБ'ЄКТАХ АБО КОЛЕКЦІЯХ ГЕОГРАФІЧНИХ ОБ'ЄКТІВ.....	54
10	GML СХЕМА– ГЕОМЕТРИЧНІ ПРИМІТИВИ.....	55
10.1	ЗАГАЛЬНІ КОНЦЕПЦІЇ.....	55
10.1.1	Огляд.....	55
10.1.2	Зв'язок з ISO 19107.....	55
10.1.3	Абстрактна геометрія.....	56
10.1.4	Координатна геометрія, вектори та обмежувальні прямокутники.....	58
10.2	АБСТРАКТНІ ГЕОМЕТРИЧНІ ПРИМІТИВИ.....	61
10.3	ГЕОМЕТРИЧНІ ПРИМІТИВИ (НУЛЬВИМІРНІ).....	62
10.4	ГЕОМЕТРИЧНІ ПРИМІТИВИ (ОДНОВИМІРНІ).....	63
10.4.1	AbstractCurveType, AbstractCurve.....	63
10.4.2	CurvePropertyType, curveProperty.....	64
10.4.3	CurveArrayPropertyType, curveArrayProperty.....	64
10.4.4	LineStringType, LineString.....	65
10.4.5	CurveType, Curve.....	65
10.4.6	OrientableCurveType, OrientableCurve, baseCurve.....	66
10.4.7	Криволінійні відрізки.....	66
10.5	ГЕОМЕТРИЧНІ ПРИМІТИВИ (ДВОХВИМІРНІ).....	78
10.5.1	AbstractSurfaceType, AbstractSurface.....	78
10.5.2	SurfacePropertyType, surfaceProperty.....	78
10.5.3	SurfaceArrayPropertyType, surfaceArrayProperty.....	79
10.5.4	PolygonType, Polygon.....	79
10.5.5	exterior, interior.....	80
10.5.6	AbstractRingType, AbstractRing.....	80
10.5.7	AbstractRingPropertyType.....	80
10.5.8	LinearRingType, LinearRing.....	80
10.5.9	LinearRingPropertyType.....	81
10.5.10	SurfaceType, Surface.....	81
10.5.11	OrientableSurfaceType, OrientableSurface, baseSurface.....	81
10.5.12	Ділянки поверхні.....	84
10.6	ТРЬОХВИМІРНІ ГЕОМЕТРИЧНІ ПРИМІТИВИ.....	89
10.6.1	AbstractSolidType, AbstractSolid.....	89
10.6.2	SolidPropertyType, solidProperty.....	89

11 GML СХЕМА – ГЕОМЕТРИЧНИЙ КОМПЛЕКС, ГЕОМЕТРИЧНІ КОМПОЗИТИ ТА ГЕОМЕТРИЧНІ АГРЕГАТИ .....	91
11.1 ЗАГАЛЬНИЙ ОГЛЯД.....	91
11.2 ГЕОМЕТРИЧНІ КОМПЛЕКСИ ТА ГЕОМЕТРИЧНІ КОМПОЗИТИ .....	92
11.2.1 Геометричний комплекс .....	92
11.2.2 Геометрії композиту .....	93
11.3 ГЕОМЕТРИЧНІ АГРЕГАТИ .....	95
11.3.1 Агрегати невизначеної розмірності .....	95
11.3.2 Нульвимірні агрегати .....	97
11.3.3 Одновимірні агрегати.....	98
11.3.4 Двохвимірні агрегати .....	99
11.3.5 Трьохвимірні агрегати .....	100
12 GML СХЕМА – СХЕМИ РЕФЕРЕНЦНИХ СИСТЕМ КООРДИНАТ .....	101
12.1 ОГЛЯД .....	101
12.1.1 Вступ.....	101
12.1.2 Зв’язок з ISO 19111 .....	101
12.1.3 Важливі XML елементи.....	102
12.2 РЕФЕРЕНЦНІ СИСТЕМИ.....	102
12.2.1 ОГЛЯД.....	102
12.2.2 IdentifiedObjectType .....	103
12.2.3 Абстрактна референцна система координат .....	103
12.3 РЕФЕРЕНЦНІ СИСТЕМИ КООРДИНАТ.....	104
12.3.1 ОГЛЯД.....	104
12.3.2 Абстрактні референцні системи координат .....	105
12.3.3 Конкретні референцні системи координат .....	106
12.4 СИСТЕМИ КООРДИНАТ.....	113
12.4.1 ОГЛЯД.....	113
12.4.2 Вісі системи координат.....	114
12.4.3 Абстрактна система координат .....	116
12.4.4 Конкретні координатні системи .....	117
12.5 ДАТИ .....	122
12.5.1 ОГЛЯД.....	122
12.5.2 Абстрактна дата .....	122
12.5.3 Геодезична дата.....	124
12.5.4 Інші конкретні дати .....	127
12.6 КООРДИНАТНІ ОПЕРАЦІЇ.....	129
12.6.1 ОГЛЯД.....	129
12.6.2 Абстрактні координатні операції .....	130
12.6.3 Конкретні координатні операції.....	134
12.6.4 Значення параметрів та групи.....	138
12.6.5 Метод операції.....	141
12.6.6 Параметри та групи операцій.....	142
13 GML СХЕМА – ТОПОЛОГІЯ .....	144
13.1 ЗАГАЛЬНІ КОНЦЕПЦІЇ .....	144
13.1.1 ОГЛЯД.....	144

13.1.2 Зв'язок з ISO 19107.....	145
13.2 АБСТРАКТНА ТОПОЛОГІЯ .....	145
13.3 ТОПОЛОГІЧНІ ПРИМІТИВИ .....	145
13.3.1 Абстрактні топологічні примітиви .....	145
13.3.2 Топологічні примітиви (нульвимірні) .....	146
13.3.3 Топологічні примітиви (одновимірні).....	147
13.3.4 Топологічні примітиви (двохвимірні) .....	148
13.3.5 Топологічні примітиви (трьохвимірні) .....	149
13.4 ТОПОЛОГІЧНІ КОЛЕКЦІЇ .....	151
13.4.1 Топологічна колекція (нульвимірна).....	151
13.4.2 Топологічна колекція (одновимірна).....	151
13.4.3 Топологічна колекція (двохвимірна).....	152
13.4.4 Топологічна колекція (трьохвимірна) .....	153
13.5 ТОПОЛОГІЧНИЙ КОМПЛЕКС .....	153
13.5.1 ТороComplexType, ТороComplex .....	153
13.5.2 Максимальний комплекс, підкомплекси (sub-complexes), надкомплекси (super-complexes).....	154
14 GML СХЕМА – ЧАСОВА ІНФОРМАЦІЯ ТА ДИНАМІЧНІ ОБ'ЄКТИ .....	155
14.1 ЗАГАЛЬНІ КОНЦЕПЦІЇ .....	155
14.1.1 Загальний огляд.....	155
14.1.2 Зв'язок з ISO 19108.....	156
14.2 ЧАСОВА СХЕМА.....	156
14.2.1 Абстрактні часові об'єкти .....	156
14.2.2 Часова геометрія .....	158
14.3 ЧАСОВА СХЕМА ТОПОЛОГІЇ.....	165
14.3.1 Вступ .....	165
14.3.2 Об'єкти часової топології.....	165
14.4 ЧАСОВІ РЕФЕРЕНЦІНІ СИСТЕМИ .....	169
14.4.1 Огляд.....	169
14.4.2 Базові часові референцні системи, TimeReferenceSystem.....	169
14.4.3 TimeCoordinateSystem .....	170
14.4.4 Календарі та годинники.....	171
14.4.5 Часові порядкові референцні системи.....	174
14.5 ПОДАННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ .....	177
14.5.1 Огляд.....	177
14.5.2 DataSource .....	177
14.5.3 Динамічні властивості.....	177
14.5.4 DynamicFeature .....	178
14.5.5 DynamicFeatureCollection.....	178
15 GML СХЕМА – ВИЗНАЧЕННЯ ТА СЛОВНИКИ .....	182
15.1 ОГЛЯД .....	182
15.2 СХЕМА СЛОВНИКА.....	182
15.2.1 Definition, DefinitionType, remarks .....	182
15.2.2 Dictionary, DictionaryType.....	183
15.2.3 dictionaryEntry, definitionMember, DictionaryEntryType.....	184

15.2.4 Використання визначень та словників .....	185
16 GML СХЕМА – ОДИНИЦІ, МІРИ ТА ЗНАЧЕННЯ .....	186
16.1 ВСТУП .....	186
16.2 СХЕМА ОДИНИЦЬ .....	186
16.2.1 Огляд.....	186
16.2.2 Використання визначень одиниць .....	187
16.2.3 unitOfMeasure, UnitOfMeasureType.....	187
16.2.4 UnitDefinition, UnitDefinitionType .....	187
16.2.5 quantityType, quantityTypeReference .....	188
16.2.6 catalogSymbol.....	188
16.2.7 BaseUnit, BaseUnitType, unitsSystem.....	188
16.2.8 DerivedUnit, DerivedUnitType .....	189
16.2.9 derivationUnitTerms, DerivationUnitTermType .....	189
16.2.10 ConventionalUnit, ConventionalUnitType .....	190
16.2.11 conversionToPreferredUnit, roughConversionToPreferredUnit, .....	190
ConversionToPreferredUnitType, FormulaType .....	190
16.2.12 Приклади словника одиниць (довідкові) .....	191
16.3 СХЕМА МІР .....	193
16.3.1 Огляд.....	193
16.3.2 measure .....	193
16.3.3 Скалярні типи міри.....	193
16.3.4 angle.....	194
16.4 СХЕМА ОБ’ЄКТІВ ЗНАЧЕННЯ .....	194
16.4.1 Вступ .....	194
16.4.2 Ієрархія елементів значень .....	195
16.4.3 Boolean, BooleanList .....	195
16.4.4 Category, CategoryList.....	196
16.4.5 Count, CountList.....	196
16.4.6 Quantity, QuantityList .....	197
16.4.7 AbstractValue, AbstractScalarValue, AbstractScalarValueList .....	198
16.4.8 Value .....	198
16.4.9 valueProperty, valueComponent, valueComponents.....	198
16.4.10 CompositeValue.....	199
16.4.11 ValueArray .....	200
16.4.12 Typed ValueExtents: CategoryExtent, CountExtent, QuantityExtent.....	202
16.4.13 BooleanPropertyType, CategoryPropertyType, CountPropertyType, QuantityPropertyType.....	203
17 GML СХЕМА – НАПРЯМКИ .....	203
17.1 СХЕМА НАПРЯМКІВ.....	203
17.2 direction, DirectionPropertyType .....	203
17.3 DirectionVectorType .....	204
17.4 DirectionDescriptionType .....	204
18 GML СХЕМА – СПОСТЕРЕЖЕННЯ .....	206
18.1 СПОСТЕРЕЖЕННЯ .....	206
18.2 СХЕМА СПОСТЕРЕЖЕННЯ .....	206

18.2.1	Огляд.....	206
18.2.2	Observation .....	206
18.2.3	using.....	207
18.2.4	target .....	208
18.2.5	resultOf .....	209
18.2.6	DirectedObservation .....	209
18.2.7	DirectedObservationAtDistance .....	210
19	GML СХЕМА – ПОКРИТТЯ.....	211
19.1	МОДЕЛЬ ТА ПОДАННЯ ПОКРИТТЯ .....	211
19.1.1	Загальні зауваження .....	211
19.1.2	Формальний опис покриття .....	212
19.1.3	Покриття в GML .....	212
19.1.4	Зв’язок з ISO 19123.....	213
19.2	СХЕМА МЕРЕЖ.....	213
19.2.1	Огляд.....	213
19.2.2	Grid .....	213
19.2.3	RectifiedGrid.....	215
19.3	СХЕМА ПОКРИТТЯ.....	216
19.3.1	AbstractCoverageType, AbstractCoverage .....	216
19.3.2	AbstractDiscreteCoverageType, AbstractDiscreteCoverage .....	217
19.3.3	AbstractContinuousCoverageType, AbstractContinuousCoverage.....	218
19.3.4	domainSet, DomainSetType.....	218
19.3.5	rangeSet, RangeSetType.....	219
19.3.6	DataBlock.....	219
19.3.7	rangeParameters .....	220
19.3.8	tupleList.....	220
19.3.9	doubleOrNilReasonTupleList .....	220
19.3.10	File, FileType .....	221
19.3.11	coverageFunction, CoverageFunctionType .....	222
19.3.12	CoverageMappingRule, MappingRule.....	224
19.3.13	GridFunction, GridFunctionType .....	224
19.3.14	sequenceRule, SequenceRuleType, SequenceRuleEnumeration.....	225
19.3.15	Особливі типи покриття у GML.....	226
19.3.16	MultiPointCoverage.....	226
19.3.17	MultiCurveCoverage .....	228
19.3.18	MultiSurfaceCoverage.....	229
19.3.19	MultiSolidCoverage.....	231
19.3.20	GridCoverage .....	231
19.3.21	RectifiedGridCoverage.....	232
20	ПРОФІЛІ .....	233
20.1	GML ПРОФІЛІ ТА ПРИКЛАДНІ СХЕМИ.....	233
20.2	ВИЗНАЧЕННЯ ПРОФІЛЮ .....	234
20.3	ЗВ’ЯЗОК З ПРИКЛАДНОЮ СХЕМОЮ.....	234
20.4	ПРАВИЛА ДЛЯ ЕЛЕМЕНТІВ ТА ТИПІВ ПРОФІЛЮ .....	234
20.5	ПРАВИЛА ПОСИЛАНЬ НА GML ПРОФІЛІ З ПРИКЛАДНИХ СХЕМ .....	235



20.6 РЕКОМЕНДАЦІ ДЛЯ ПРИКЛАДНИХ СХЕМ З ВИКОРИСТАННЯМ GML ПРОФІЛІВ .....	236
20.7 Підсумок правил для GML ПРОФІЛІВ .....	236
21 ПРАВИЛА ДЛЯ GML ПРИКЛАДНИХ СХЕМ.....	237
21.1 GML ЕКЗЕМПЛЯРИ ОБ'ЄКТІВ .....	237
21.1.1 GML документи .....	237
21.1.2 GML елементи об'єкту в інших XML документах .....	237
21.2 GML ПРИКЛАДНІ СХЕМИ.....	238
21.2.1 Вступ .....	238
21.2.2 Цільовий простір імен.....	239
21.2.3 Імпорт GML схеми.....	239
21.2.4 Отримання типу об'єкту .....	240
21.2.5 Елементи, якими подають об'єкти.....	240
21.2.6 Отримання типу властивості.....	240
21.2.7 Елементи, якими подають властивості.....	240
21.3 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ГЕОГРАФІЧНІ ОБ'ЄКТИ ТА КОЛЕКЦІЇ ГЕОГРАФІЧНИХ ОБ'ЄКТІВ.....	241
21.3.1 Вступ.....	241
21.3.2 Компоненти GML схеми імпорту .....	241
21.3.3 Елементи, якими подають об'єкти.....	241
21.3.4 Географічні об'єкти аплікації є географічними об'єктами.....	242
21.4 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ПРОСТОРОВІ ГЕОМЕТРІЇ .....	242
21.4.1 Компоненти імпорту GML схеми геометрії.....	242
21.4.2 Типи геометрії та типи властивостей геометрії, визначені користувачем.....	242
21.5 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ПРОСТОРОВІ ТОПОЛОГІЇ.....	243
21.5.1 Компоненти GML схеми імпорту топології.....	243
21.5.2 Визначені користувачем типи топології та властивостей топології .....	244
21.6 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ЧАС .....	244
21.6.1 Компоненти GML часової схеми імпорту .....	244
21.6.2 Часові типи та типи часової властивості, визначені користувачем.....	244
21.7 СХЕМИ, ЩО ВИЗНАЧАЮТЬ РЕФЕРЕНЦІНІ СИСТЕМИ КООРДИНАТ .....	245
21.7.1 Вступ .....	245
21.7.2 Імпорт компонентів GML схеми референційної системи координат.....	246
21.8 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ПОКРИТТЯ .....	246
21.8.1 Вступ.....	246
21.8.2 Імпорт компоненти GML схеми покриття.....	246
21.8.3 Типи покриттів, визначені користувачем .....	247
21.8.4 Параметри діапазону мають бути здатними замінювати AbstractValue .....	247
21.8.5 Документ покриття.....	248
21.9 СХЕМИ, ЩО ВИЗНАЧАЮТЬ СПОСТЕРЕЖЕННЯ .....	248
21.9.1 Вступ.....	248
21.9.2 Імпорт компонентів GML схеми спостереження .....	248
21.9.3 Типи спостережень, визначені користувачем .....	248
21.9.4 Колекції спостережень .....	249
21.9.5 Спостереження як географічні об'єкти .....	249
21.9.6 Документ колекції спостережень .....	249

21.10 СХЕМИ, ЩО ВИЗНАЧАЮТЬ СЛОВНИКИ .....	249
21.10.1 Вступ.....	249
21.10.2 Компоненти імпорту словника GML схеми .....	249
21.10.3 Типи визначень, визначені користувачем .....	249
21.10.4 Визначені користувачем типи словників.....	249
21.11 СХЕМИ, ЩО ВИЗНАЧАЮТЬ ЗНАЧЕННЯ.....	250
21.11.1 Вступ.....	250
21.11.2 Компоненти імпорту GML схеми об'єктів значень .....	250
21.11.3 Побудова нових типів значень .....	250
21.12 GML ПРОФІЛІ GML СХЕМИ .....	250
ДОДАТОК А.....	252
(НОРМАТИВНИЙ) .....	252
АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ GML ПРИКЛАДНИХ СХЕМИ, GML ПРОФІЛІВ ТА GML ДОКУМЕНТІВ .....	252
А.1 АБСТРАКТНІ КОМПЛЕКТИ ТЕСТІВ ДЛЯ GML ПРИКЛАДНИХ СХЕМ.....	252
А.1.1 Випадки тестів для обов'язкових класів відповідності .....	252
А.1.2 Випадки тестів GML прикладних схем, конвертованих з прикладних схем стандарту ISO 19109 в UML .....	255
А.1.3 Випадки тестів прикладних схем стандарту ISO 19109 в UML, конвертованих з GML прикладних схем .....	255
А.1.4 GML прикладні схеми, що визначають географічні об'єкти та колекції географічних об'єктів .....	256
А.1.5 GML прикладні схеми, що визначають просторові геометрії.....	256
А.1.6 GML прикладні схеми, що визначають просторові топології.....	256
А.1.7 GML прикладні схеми, що визначають час .....	256
А.1.8 GML прикладні схеми, що визначають референсні системи координат .....	256
А.1.9 GML прикладні схеми, що визначають покриття.....	257
А.1.10 GML прикладні схеми, що визначають спостереження.....	257
А.1.11 GML прикладні схеми, що визначають словники та визначення.....	257
А.1.12 GML прикладні схеми, що визначають значення.....	257
А.2 АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ GML ПРОФІЛІВ: .....	257
А.2.1 Дійсний GML профіль.....	257
А.2.2 Геометричні примітиви (просторові).....	257
А.2.3 Геометричні комплекси (просторові) .....	258
А.2.4 Топологічні комплекси (просторові) .....	259
А.2.5 Топологічні комплекси з геометричною реалізацією (просторові).....	260
А.2.6 Референсні системи координат .....	260
А.2.7 Координатні операції.....	260
А.2.8 Часова геометрія .....	260
А.2.9 Часова топологія .....	261
А.2.10 Часові референсні системи .....	261
А.2.11 Динамічні географічні об'єкти .....	261
А.2.12 Словники .....	261
А.2.13 Словники одиниць .....	262
А.2.14 Спостереження .....	262

A.2.15 Покриття.....	262
A.3 АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ GML ДОКУМЕНТІВ .....	263
A.3.1 Існування посилання на застосовну GML прикладну схему .....	263
A.3.2 Існування GML прикладної схеми, на яку спрямоване посилання .....	263
A.3.3 Відповідність GML прикладної схеми, на яку спрямоване посилання.....	264
A.3.4 Дійсний XML.....	264
A.3.5 Відповідність GML документу .....	264
ДОДАТОК Б .....	264
(НОРМАТИВНИЙ) .....	264
АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	264
B.1 Випадки тестів на дотримання обов’язкових вимог відповідності.....	264
B.1.1 GML профіль .....	264
B.1.2 Підтримка локальних простих Xlink посилань.....	265
B.1.3 Референцні системи координат, які використовують у географічних об’єктах (реалізація в програмному забезпеченні).....	265
B.2 Випадки тестів дотримання необов’язкових вимог відповідності для РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІЗ ЗДАТНІСТЮ ОБРОБЛЯТИ GML ОБ’ЄКТИ У XML ФОРМАТІ.....	265
B.2.1 Підтримка віддалених простих Xlink .....	265
B.2.2 Підтримка розширених Xlink посилань.....	265
B.2.3 Підтримка для властивостей, які можна обнуляти .....	266
B.2.4 Підтримка одиниць вимірювання .....	266
B.2.5 Підтримка семантики власності властивостей .....	266
B.2.6 Властивості метаданих .....	266
B.2.7 Підтримка GML профілів в перевірці екземпляру.....	266
B.3 Випадки тестів для написання GML .....	266
B.3.1 Можливість серіалізації.....	266
B.3.2 Дійсність серіалізації .....	267
B.4 Випадок тесту читання GML .....	267
B.5 Випадки тестів для написання GML ПРИКЛАДНИХ СХЕМ.....	267
B.5.1 Здатність на серіалізацію.....	267
B.5.2 Дійсність серіалізації .....	267
B.6 Випадки тестів для читання GML прикладних схем .....	267
ДОДАТОК В.....	268
(ДОВІДКОВИЙ) .....	268
GML СХЕМА .....	268
ДОДАТОК Г .....	269
(НОРМАТИВНИЙ) .....	269
РЕАЛІЗОВАНИЙ ПРОФІЛЬ МІЖНАРОДНИХ СТАНДАРТІВ КОМПЛЕКСУ ISO 19100 ТА РОЗШИРЕНЬ .....	269
Г.1 ЗАГАЛЬНІ ЗАУВАЖЕННЯ .....	269
Г.2 ПРОФІЛЬ КОМПЛЕКСУ ISO 19100, ВИКОРИСТАНИЙ У GML.....	269
Г.2.1 Огляд.....	269
Г.2.2 Мова концептуальної схеми ISO/TS 19103 .....	277

Г.2.3 Просторова схема (геометрія) стандарту ISO 19107 .....	278
Г.2.4 Просторова схема (топология) стандарту ISO 19107 .....	298
Г.2.5 Часова схема ISO 19108.....	305
Г.2.6 Правила ISO 19109 для прикладної схеми .....	313
Г.2.7 Просторова прив'язка по координатах відповідно до ISO 19111 .....	314
Г.2.8 Просторова прив'язка по географічних ідентифікаторах ISO 19112.....	316
Г.2.9 Метадані ISO 19115.....	316
Г.2.10 Кодування ISO 19118.....	316
Г.2.11 Покриття ISO 19123 .....	316
Г.3 РОЗШИРЕННЯ ПРОФІЛЮ КОМПЛЕКСУ МІЖНАРОДНИХ СТАНДАРТІВ 19100.....	319
Г.3.1 Огляд.....	319
Г.3.2 Пакет “basicTypes” .....	319
Г.3.3 Пакет “gmlBase” .....	322
Г.3.4 Пакет “географічний об’єкт“ (“feature”).....	322
Г.3.5 Пакет “geometryBasic0d01” .....	323
Г.3.6 Пакет “geometryBasic2d” .....	324
Г.3.7 Пакет “geometryPrimitives” .....	325
Г.3.8 Пакет “geometryAggregates” .....	325
Г.3.9 Пакети “coordinateOperations”, “coordinateReferenceSystems”, .....	326
“coordinateSystems”, “dataQuality”, “datums”, “referenceSystems” .....	326
Г.3.10 Пакет “topology” .....	330
Г.3.11 Пакет “dynamicFeature” .....	331
Г.3.12 Пакет “dictionary” (словник).....	332
Г.3.13 Пакет “units” (одиниці).....	333
Г.3.14 Пакет “measures” (міри).....	333
Г.3.15 Пакет “valueObjects” .....	334
Г.3.16 Пакет “direction” (напрямок) .....	337
Г.3.17 Пакет “observation” (спостереження) .....	338
ДОДАТОК Д.....	338
(НОРМАТИВНИЙ) .....	338
ПРАВИЛА КОДУВАННЯ ПЕРЕХОДУ З UML ДО GML ПРИКЛАДНОЇ СХЕМИ .....	338
Д.1 ЗАГАЛЬНІ КОНЦЕПЦІЇ .....	338
Д.2 ПРАВИЛА КОДУВАННЯ.....	339
Д.2.1 Загальні вимоги до кодування.....	339
Д.2.2 Структура вхідних даних .....	344
Д.3 ПРИКЛАД <ДОВІДКОВИЙ> .....	358
ДОДАТОК Е.....	361
(НОРМАТИВНИЙ) .....	361
ПРАВИЛА КОДУВАННЯ ПЕРЕХОДУ ВІД GML ДО UML ПРИКЛАДНОЇ СХЕМИ ..	361
Е.1 ЗАГАЛЬНІ КОНЦЕПЦІЇ.....	361
Е.2 ПРАВИЛА КОДУВАННЯ .....	362
Е.2.1 Загальні вимоги до кодування.....	362
Е.2.2 Структура даних на виході .....	366
Е.2.3 Правила перетворення .....	366
ДОДАТОК Ж.....	371

(ДОВІДКОВИЙ) .....	371
ІНСТРУКЦІЇ З ПОДІЛУ GML СХЕМИ НА ПІДНАБОРИ .....	371
Ж.1 ЗАГАЛЬНІ ВІДОМОСТІ .....	371
Ж.2 DEPENDS.XSLT .....	371
Ж.3 GMLSUBSET.XSLT .....	378
Ж.4 UTILITY.XSLT .....	382
ДОДАТОК И.....	387
(ДОВІДКОВИЙ) .....	387
СТИЛІЗАЦІЯ ЗА УМОВЧАННЯМ .....	387
И.1 ЗАГАЛЬНІ ВІДОМОСТІ .....	387
И.2 ЕЛЕМЕНТИ СТИЛІЗАЦІЇ ВИЩОГО РІВНЯ .....	388
И.2.1 Огляд .....	388
И.2.2 defaultStyle .....	388
И.2.3 Стиль .....	389
И.3 СТИЛЬ ГЕОГРАФІЧНОГО ОБ'ЄКТУ .....	390
И.3.1 FeatureStyle .....	390
И.3.2 featureType .....	391
И.3.3 baseType .....	391
И.3.4 featureConstraint .....	391
И.3.5 queryGrammar .....	391
И.4 СТИЛЬ ГЕОМЕТРІЇ .....	391
И.5 СТИЛЬ ТОПОЛОГІЇ .....	392
И.6 СТИЛЬ НАПИСІВ .....	393
И.7 ЗАГАЛЬНІ ЕЛЕМЕНТИ СТИЛІЗАЦІЇ .....	395
И.7.1 Огляд .....	395
И.7.2 symbol .....	395
И.7.3 styleVariation .....	395
И.7.4 spatialResolution .....	396
И.7.5 animation .....	397
И.8 СТИЛЬ ГРАФІВ .....	397
ДОДАТОК К.....	400
(ДОВІДКОВИЙ) .....	400
ЗВОРОТНА СУМІСНІСТЬ З ПОПЕРЕДНІМИ ВЕРСІЯМИ GML .....	400
К.1 Огляд .....	400
К. 2 БАЗОВІ КОМПОНЕНТИ СХЕМИ .....	400
К.2.1 remoteSchema .....	400
К.2.2 member .....	400
К.2.3 ArrayAssociationType .....	400
К.2.4 members .....	400
К.2.5 featureProperty, featureMember, featureMembers .....	400
К.2.6 StringOrRefType .....	401
К.2.7 Array, ArrayType, Bag, BagType .....	401
К.2.8 metaDataProperty, MetaDataPropertyType, AbstractMetaData, AbstractMetaDataType .....	402
К.2.9 GenericMetaData, GenericMetaDataType .....	402

К.3 BASIC TYPES, NULL.....	403
К. 4 ГЕОГРАФІЧНІ ОБ'ЄКТИ.....	404
К.4.1 location, LocationPropertyType, LocationKeyWord, LocationString.....	404
К.4.2 priorityLocation, priorityLocationType .....	404
К.4.3 BoundedFeatureType .....	405
К.4.4 AbstractFeatureCollectionType, AbstractFeatureCollection, FeatureCollection, FeatureCollectionType .....	405
К.4.5 Просторові властивості .....	406
К.5 КООРДИНАТНА ГЕОМЕТРІЯ, ГЕОМЕТРИЧНІ ПРИМІТИВИ.....	407
К.5.1 coordinates .....	407
К.5.2 pos у EnvelopeType.....	407
К.5.3 pointRep.....	407
К.5.4 polygonPatches.....	407
К.5.5 trianglePatches.....	407
К.6 РЕФЕРЕНЦІНІ СИСТЕМИ КООРДИНАТ .....	408
К.6.1 baseGeographicCRS .....	408
К.6.2 GeographicCRS.....	408
К.6.3 GeocentricCRS .....	408
К.6.4 uom.....	409
К.6.5 ObliqueCartesianCS.....	409
К.6.6 TemporalCS.....	410
К.6.7 greenwichLongitude.....	410
К.6.8 AbstractOperation .....	410
К.6.9 dmsAngleValue .....	411
К.6.10 Перейменовані елементи властивості.....	411
К.6.11 Ref елементи властивості.....	412
К.7 ЧАСОВА ІНФОРМАЦІЯ ТА ДИНАМІЧНІ ОБ'ЄКТИ .....	413
К.7.1 SuccessionType .....	413
К.7.2 MovingObjectStatus.....	414
К.7.3 track.....	414
К.8 ВИЗНАЧЕННЯ ТА СЛОВНИКИ.....	414
К.8.1 DefinitionCollection.....	414
К.8.2 definitionMember.....	415
К.8.3 indirectEntry, IndirectEntryType, DefinitionProxy, DefinitionProxyType .....	415
К.9 ОДИНИЦІ, МІРИ ТА ЗНАЧЕННЯ .....	416
К.9.1 dmsAngle .....	416
К.9.2 degrees .....	417
К.9.3 decimalMinutes .....	417
К.9.4 minutes .....	417
К.9.5 seconds .....	417
К.9.6 AngleChoiceType.....	418
К.10 НАПРЯМКИ.....	418
К.11 ПОКРИТТЯ .....	418
К.11.1 MappingRule .....	418
К.11.2 IncrementOrder .....	418

К.11.3 Властивості набору доменів .....	418
ДОДАТОК Л.....	419
(ДОВІДКОВИЙ) .....	419
МОДУЛЯРИЗАЦІЯ ТА ЗАЛЕЖНОСТІ .....	419
БІБЛІОГРАФІЯ .....	421

## НАЦІОНАЛЬНИЙ ВСТУП

Цей стандарт входить до комплексу міжнародних стандартів ISO 19100 “Географічна інформація / геоматика”, які визначають моделі географічних даних, адміністрування цих даних та геоінформаційні сервіси.

Цей стандарт встановлює загальні вимоги, принципи розроблення та базові XML-схеми для обмінних форматів геопросторових даних на основі географічної мови розмітки GML.

Технічний комітет, відповідальний за цей стандарт, – ТК 103 “Географічна інформація / геоматика”.

До стандарту внесено такі редакційні зміни:

- з тексту вилучено попередній довідковий матеріал ISO 19136:2007;
- слова “Міжнародний стандарт” замінено словами “Цей стандарт”;
- структурні елементи стандарту – “Обкладинка”, “Передмова”, “Національний вступ” та “Бібліографічні дані” – оформлено згідно з вимогами державної системи стандартизації; - терміни стандарту при перекладі уніфіковано і враховано стандарти системи оброблення інформації.

Для пояснення окремих національних особливостей застосування стандарту в ньому курсивом подано “Національні примітки” до розділів \_.



СТАНДАРТ МІНІСТЕРСТВА ОХОРОНИ НАВКОЛИШНЬОГО  
ПРИРОДНОГО СЕРЕДОВИЩА УКРАЇНИ

---

**ОЛДОЛОЛОЛОЛОЛОЛДОЛОЛДОЛДОЛДОЛОЛОЛОЛОЛДОЛОЛОЛОЛОЛОЛОЛ**

---

КОМПЛЕКС СТАНДАРТІВ  
“ГЕОГРАФІЧНА ІНФОРМАЦІЯ / ГЕОМАТИКА”  
ГЕОГРАФІЧНА ІНФОРМАЦІЯ – ГЕОГРАФІЧНА МОВА РОЗМІТКИ (GML)

КОМПЛЕКС СТАНДАРТОВ  
“ГЕОГРАФИЧЕСКАЯ ИНФОРМАЦИЯ / ГЕОМАТИКА”  
ГЕОГРАФИЧЕСКАЯ ИНФОРМАЦИЯ – ЯЗЫК ГЕОГРАФИЧЕСКОЙ РАЗМЕТКИ (GML)

SERIES OF GEOGRAPHIC INFORMATION STANDARDS  
“GEOGRAPHIC INFORMATION / GEOMATICS”  
GEOGRAPHIC INFORMATION — GEOGRAPHY MARKUP LANGUAGE (GML)

---

Чинний від \*\*. \*\*200\*

## ПЕРЕДМОВА

Міжнародна організація по стандартизації (ISO) – це всесвітня федерація національних організацій з державної стандартизації (організацій-учасників ISO).

Як правило, роботу з підготовки міжнародних стандартів виконують через технічні комітети ISO. Кожна організація, що працює в сфері діяльності, для якої був створений технічний комітет, має право бути представленою у ньому. Міжнародні, урядові і неурядові організації, які зв’язані з ISO, також беруть участь у роботі технічного комітету. ISO тісно співпрацює з Міжнародною Електротехнічною Комісією (IEC) із усіх питань, що стосуються електротехнічних стандартів.

Міжнародні стандарти розробляють відповідно до правил, що викладені у Директивах ISO/IEC, Частина 2.

Основна мета роботи технічних комітетів полягає у підготовці міжнародних стандартів. Проекти міжнародних стандартів, прийняті технічними комітетами, поширюють серед національних організацій-членів ISO для голосування. Проект, схвалений щонайменше 75 % голосів організацій-членів, публікують як міжнародний стандарт.

Слід зауважити, що деякі елементи цього стандарту можуть бути предметом патентних прав. ISO не відповідає за виявлення будь-якого або усіх таких патентних прав.

Географічна мова розмітки (GML) почала свій розвиток у рамках Відкритого геопросорового консорціуму (OGC). ISO 19136 був підготовлений Технічним комітетом ISO/TC 211, Geographic information/Geomatics, спільно з консорціумом OGC.

## ВСТУП

Географічна мова розмітки є XML граматиною, описаною в XML схемах для подання прикладних схем як для передавання так і для зберігання географічної інформації.

Ключові концепції, що використовуються у географічній мові розмітки (GML) для моделювання світу, запозичені з комплексу міжнародних стандартів ISO 19100 та Абстрактної специфікації OpenGIS.

Об'єкт є абстракцією явищ реального світу (ISO 19101); він є географічним об'єктом, якщо асоційований із місцеположенням відносно Землі. Таким чином, цифрове подання реального світу можна відображати як набір об'єктів. Стан об'єкту визначається набором властивостей, в якому кожна властивість подається трійкою {ім'я, тип, значення}.

Множина властивостей, що можуть належати об'єкту, разом із їх іменами та типами, визначається через визначення типу. Географічні об'єкти з геометрією – такі, що мають геометрично визначені властивості. Колекція географічних об'єктів є колекцією об'єктів, яку можна також вважати об'єктом; з цього випливає, що колекція об'єктів має тип об'єкту, а отже і свої властивості, в доповнення об'єктам, які вона містить.

Згідно з ISO 19109, типи об'єктів прикладної сфери або її домену зазвичай входять до прикладної схеми. GML прикладна схема визначена в XML схемі та може бути побудована двома альтернативними способами:

- через дотримання правил для прикладних схем у UML, згідно з ISO 19109, та у відповідності з обмеженнями, накладеними на такі схеми, так і правилами їх відображення в GML прикладні схеми, що визначені у цьому стандарті;
- з дотриманням правил для GML прикладних схем, визначених у цьому стандарті, що описує правила розробки GML прикладних схем безпосередньо в XML схемах.

Обидва шляхи підтримуються в цьому стандарті. Для забезпечення належного користування структурою концептуального моделювання комплексу міжнародних стандартів ISO 19100 всі прикладні схеми треба моделювати у відповідності до загальної моделі географічного об'єкту, як визначено в ISO 19109. У комплексі міжнародних стандартів ISO 19100 UML є переважною мовою моделювання концептуальних схем.

GML визначає XML кодування, узгоджене з ISO 19118, для декількох концептуальних класів, визначених в комплексі міжнародних стандартів ISO 19100 та в Абстрактній специфікації OpenGIS. Ці концептуальні моделі включають такі, що визначені в:

- SO/TS 19103 — Conceptual schema language (units of measure, basic types) (Мова концептуальних схем (одиниці вимірів, базові типи);
- ISO 19107 — *Spatial schema (geometry and topology objects)* — Просторова схема (геометричні і топологічні об'єкти);
- ISO 19108 — *Temporal schema (temporal geometry and topology objects, temporal reference systems)* — Часова схема (часові геометричні і топологічні об'єкти, часові системи відліку);
- ISO 19109 — *Rules for application schemas (features)* — Правила для прикладних схем (об'єкти);

- ISO 19111 — *Spatial referencing by coordinates (coordinate reference systems)* — Просторова прив'язка за координатами (референцні системи координат);
- ISO 19123 — *Schema for coverage geometry and functions* — Схеми для геометрії та функцій покриття.

Метою є забезпечення стандартизованого кодування (реалізація в XML) типів, визначених у концептуальних моделях вище згаданих міжнародних стандартів. Якщо кожна прикладна схема закодована незалежно а в процес кодування включені однакові типи, наприклад, з ISO 19108, то тоді без однозначних та повністю фіксованих правил XML кодування будуть різними. Крім того, оскільки кожна платформа реалізації має свої недоліки та переваги, то доцільно стандартизувати XML кодування для базових концепцій географічної інформації, що моделюються в комплексі міжнародних стандартів ISO 19100 та широко використовуються в прикладних схемах.

Переважно відображення концептуальних класів є прямим, але іноді воно складніше (детальний опис відображення є частиною цього стандарту).

Крім того, GML надає XML кодування для додаткових концепцій, ще не змодельованих в комплексі міжнародних стандартів ISO 19100 або в Абстрактній специфікації OpenGIS, наприклад, динамічні об'єкти, прості спостереження або об'єкти-значення.

Наперед визначені типи географічних об'єктів у GML включають покриття та прості спостереження.

Покриття є підтипом об'єкту, що має функцію покриття із просторово-часовим доменом та діапазоном множини значень однорідних кортежів із розмірністю від 1 до  $n$ . Покриття може складатись із одного об'єкту або бути колекцією об'єктів "для моделювання та зображення просторових відношень між об'єктами земної поверхні та їх просторового розподілу" (OGC Abstract Specification, Тема 6) і покриття "діє як функція для повернення значень в заданому діапазоні для кожної точки в межах його просторово-часового домену.

Спостереження моделює факт спостереження, часто камерою або іншим засобом, особою або певним інструментом (словник "Merriam-Webster Dictionary": "акт ідентифікації та фіксації факту або випадку, часто із допомогою інструментальних вимірювань"). Спостереження може вважатися GML об'єктом із моментом часу, в який відбулося спостереження, та із значенням для спостереження.

Система відліку (референцна система) забезпечує масштаб вимірювання для надання значень позиції, часу або порядку чи іншим описовим кількостям та якостям.

Референцна система координат складається із набору координатних осей, що пов'язані із Землею через дату, на яку описується розмір та форма Землі.

Система відліку часу забезпечує стандартні одиниці для вимірювання часу та опису часової протяжності чи тривалості.

Каталог референцних систем забезпечує визначення референцних систем, що використовуються у просторовій чи часовій геометрії.

Просторові геометричні елементи є значеннями просторових властивостей об'єктів. Вони ідентифікують референцну систему координат, у якій вони були виміряні. Батьківські геометричні елементи геометричний комплекс або геометричне агрегування призначені для подання складених геометричних елементів.

Часові геометричні елементи є значеннями часових властивостей об'єктів. Подібно до просторових геометричних елементів, часові геометричні елементи ідентифікують часову систему відліку, в якій були виконані вимірювання.

Просторові або часові топології використовують для подання різних топологічних відношень між об'єктами.

Каталог одиниць вимірювань забезпечує визначення числових вимірювань та фізичних величин, таких як довжина, температура, тиск, а також перетворення одних одиниць в інші та перетворення між одиницями.

## 1 СФЕРА ЗАСТОСОВНОСТІ

Географічна мова розмітки (GML) є XML кодуванням, що відповідає ISO 19118, для передавання і зберігання географічної інформації, змодельованої відповідно до структури концептуального моделювання, що використовується в серії Міжнародних стандартів ISO 19110 та включає як просторові, так і непросторові властивості географічних об'єктів.

Цей стандарт визначає синтаксис, механізми та угоди XML схем, які:

- забезпечують відкрити, незалежну від постачальника структуру опису геопросторових прикладних схем для передавання і зберігання географічної інформації в XML;
- передбачають профілі, що підтримують відповідні піднабори описових можливостей структури GML;
- підтримують опис геопросторових прикладних схем для спеціалізованих сфер та галузей інформатики;
- сприяють створенню та веденню зв'язаних геопросторових прикладних схем та наборів даних;
- підтримують зберігання та передавання прикладних схем та наборів даних;
- створюють умови для організації обміну географічними прикладними схемами та інформацією, яку вони описують.

Розробники можуть вирішити зберігати географічні прикладні схеми та інформацію в GML або конвертувати в деякі інші формати зберігання за вимогою та використовувати GML тільки для передавання схем і даних.

**Примітка 1.** Якщо прикладну схему, що відповідає стандарту ISO 19109 та описана в UML, використовують як основу для зберігання та передавання географічної інформації, цей стандарт визначає нормативні правила відображення такої прикладної схеми в GML прикладну схему в XML схемі, і, таким чином, в XML кодування для даних з логічною структурою, узгоджену з прикладною схемою, що відповідає стандарту ISO 19109.

## 2 ВІДПОВІДНІСТЬ

### 2.1 Вимоги до відповідності

Розділи з 7 по 19 цього міжнародного стандарту визначають компоненти XML схеми, тобто GML схему, що має бути використана в GML прикладних схемах згідно до розділу 21. Розділ 20 визначає правила для визначення GML профілю, який може бути призначений для використання в GML прикладній схемі.

Окремі аплікації вимагають повний діапазон можливостей, описаних у GML прикладній схемі. Розділ 2 таким чином визначає набір класів відповідності, що мають підтримувати аплікації, вимоги яких коливаються від мінімально необхідних для визначення простих типів об'єктів до повного використання GML схеми.

Більшість компонентів схеми, конкретизованих у цьому стандарті, визначені в комплексі Міжнародних стандартах ISO 19100. У цих випадках класи відповідності, визначені в цьому стандарті, базовані на класах відповідності, визначених у відповідному стандарті.

Будь-яка GML прикладна схема, GML профіль або реалізація програмного забезпечення, що претендує на відповідність одному з класів відповідності, має пройти всі випадки тестів відповідного абстрактного комплексу тестів.

Будь-яка реалізація програмного забезпечення, що претендує на відповідність цьому стандарту, має включати документування GML профілю, який підтримується реалізацією. GML профіль GML має пройти всі обов'язкові випадки тестування абстрактного комплекту тестів для відповідних GML профілів.

### 2.2 Класи відповідності, пов'язані із GML прикладними схемами

GML прикладні схеми, що вимагають відповідності до цього стандарту, мають задовольняти правилам, що конкретизовані в розділах з 7 по 21 та пройти всі відповідні тестові випадки абстрактного комплекту тестів в А.1.

Залежно від характеристик GML прикладної схеми виділяють 12 класів відповідності. В таблиці 1 перелічені ці класи та відповідні пункти абстрактного комплекту тестів.

**Таблиця 1** – Класи відповідності та їх зв'язок з GML прикладними схемами

Клас відповідності	Пункт абстрактного комплекту тестів
Всі GML прикладні схеми	A.1.1
GML прикладні схеми, отримані з прикладних схем ISO 19109 в UML шляхом конвертації	A.1.2
GML прикладні схеми, що мають бути отримані з прикладних схем ISO 19109 в UML шляхом конвертації	A.1.3
GML прикладні схеми, що визначають географічні об'єкти та колекції цих об'єктів	A.1.4
GML прикладні схеми, що визначають просторові геометрії	A.1.5
GML прикладні схеми, що визначають просторові топології	A.1.6
GML прикладні схеми, що визначають час	A.1.7
GML прикладні схеми, що визначають референцні системи координат	A.1.8
GML прикладні схеми, що визначають покриття	A.1.9
GML прикладні схеми, що визначають спостереження	A.1.1
GML прикладні схеми, що визначають словники і визначення	A.1.11
GML прикладні схеми, що визначають значення	A.1.12

### 2.3 Класи відповідності, зв'язані з GML профілями

Вимоги до прикладних схем визначаються в контексті компонентів XML схеми, що містяться в GML схемі, які треба включати в GML профіль. GML профілі, що вимагають відповідності до цього стандарту, мають задовольняти вимогам абстрактного комплекту тестів за А.2.

Відповідно до контенту та вимогам стосовно специфіки GML профілю, виділено 31 клас відповідності. У таблиці 2 перелічено ці класи та відповідні підкласи абстрактного комплекту тестів.

Таблиця 2 – Класи відповідності та їх зв'язок з GML профілями

Клас відповідності	Пункт абстрактного комплексу тестів
Всі профілі GML	A.2.1
Геометричні примітиви (просторові) - нульвимірні	A.2.2.1.1
Геометричні примітиви (просторові) – нуль - та одновимірні	A.2.2.1.2
Геометричні примітиви (просторові) – нуль -, одно - та двохвимірні	A.2.2.1.3
Геометричні примітиви (просторові) – нуль -, одно -, двох - та трьохвимірні	A.2.2.1.4
Геометричні комплекси (просторові) – нуль- та одновимірні	A.2.3.1.1
Геометричні комплекси (просторові) – нуль-, одно- та двохвимірні	A.2.3.1.2
Геометричні комплекси (просторові) – нуль-, одно-, двох- та трьохвимірні	A.2.3.1.3
Топологічні комплекси (просторові) – нуль- та одновимірні	A.2.4.1.1
Топологічні комплекси (просторові) – нуль-, одно- та двохвимірні	A.2.4.1.2
Топологічні комплекси (просторові) – нуль-, одно-, двох- та трьохвимірні	A.2.4.1.3
Топологічні комплекси з геометричною реалізацією (просторові) – одновимірні	A.2.5.1.1
Топологічні комплекси з геометричною реалізацією (просторові) – двохвимірні	A.2.5.1.2
Топологічні комплекси з геометричною реалізацією (просторові) – трьохвимірні	A.2.5.1.3
Референцні системи координат	A.2.6
Координатні операції між двома референцними системами координат	A.2.7
Часова геометрія – нульвимірна	A.2.8.1
Часова геометрія – нуль- та одновимірна	A.2.8.2
Часова топологія	A.2.9
Часові референцні системи	A.2.10
Динамічні об'єкти	A.2.11
Словники	A.2.12
Словники одиниць	A.2.13
Спостереження	A.2.14
Абстрактні покриття	A.2.15.1
Дискретні точкові покриття	A.2.15.2
Дискретні криволінійні покриття	A.2.15.3
Дискретні поверхневі покриття	A.2.15.4
Дискретні твердотільні покриття	A.2.15.5
GRID-покриття	A.2.15.6
Безперервні покриття	A.2.15.7

Реалізація кривих ліній для профілів GML, що включають одновимірні просторові геометричні об'єкти, має завжди включати лінійні методи інтерполяції. Реалізація поверхонь в профілях GML, що включають двохвимірні просторові геометричні об'єкти, має завжди включати планарні методи інтерполяції. Додаткові механізми інтерполяції ліній та поверхонь не є обов'язковими, але якщо їх реалізують, то вони мають відповідати визначенням, поданим у цьому стандарті.

**Примітка 1.** Доречно порівняти ці класи відповідності з ISO 19107:2003, пункт 2, ISO 19108:2002, 2.2 та ISO 19123:2005, пункт 2.

**Примітка 2.** GML профіль, що відповідає трьом класам відповідності „Геометричні примітиви (просторові) - нульвимірні”, “Геометричні примітиви (просторові) – нуль- або одновимірні”, “Геометричні примітиви (просторові) – нуль-, одно- або двохвимірні” (в додаток до класу відповідності “всі профілі GML”) має відповідати просторовому профілю, визначеному в ISO 19137:2007 та певним тестам на відповідність в ISO 19137:2007 (B.1, B.2, B.3).

#### 2.4 Класи відповідності, зв’язані з GML документами

GML документи, що відповідають цьому міжнародному стандарту, мають задовольняти правила, визначені в розділах з 7 по 21 та пройти всі відповідні тестові випадки абстрактного комплекту тестів в А.3.

#### 2.5 Класи відповідності, пов’язані із реалізацією у програмному забезпеченні

Програмне забезпечення, що підтримує читання або написання GML або GML прикладних схем та відповідає цьому Міжнародному стандарту, має пройти всі відповідні абстрактні набори тестів, описані в абстрактному наборі тестів у додатку Б.

Залежно від можливостей реалізації виділяють 11 класів відповідності. У таблиці 3 перелічені ці класи та відповідні підкласи абстрактного набору тестів.

**Таблиця 3** – Класи відповідності та їх зв’язок з реалізацією

Клас відповідності	Пункт абстрактного комплекту тестів
Всі реалізації програмного забезпечення	Б.1
Підтримка віддалених простих посилань Xlink	Б.2.1
Підтримка розширених посилань Xlink	Б.2.2
Підтримка властивостей, що можна обнулити	Б.2.3
Підтримка одиниць вимірювання	Б.2.4
Підтримка власницької семантики властивостей	Б.2.5
Властивості метаданих	Б.2.6
Підтримка профілів GML у перевірці екземпляру	Б.2.7
Написання GML	Б.3
Читання GML	Б.4
Написання GML прикладних схем	Б.5
Читання GML прикладних схем	Б.6

### 3 НОРМАТИВНІ ПОСИЛАННЯ

Нижче згадані документи, необхідні для застосування цього документу. Для датованих посилань застосовується тільки цитоване видання. Для недатованих посилань треба застосовувати найпізніше видання документу, на який посилаються (включаючи будь-які доповнення):

- ISO/IEC 11404:1996, Information technology — Programming languages, their environments and system software interfaces – Language-independent datatypes, Інформаційна технологія – Мови програмування, їх середовища та системні інтерфейси програм – Незалежні від мов типи даних;
- ISO/TS 19103:2005, Geographic Information — Conceptual schema language, Географічна інформація – Мови концептуальної схеми;



- ISO 19107:2003, Geographic Information — Spatial schema, Географічна інформація – Просторова схема;
- ISO 19108:2002, Geographic Information — Temporal schema, Географічна інформація – Часова схема;
- ISO 19109:2005, Geographic Information — Rules for application schemas, Географічна інформація – Правила для прикладних схем;
- ISO 19111:2002, — 1, Geographic Information — Spatial referencing by coordinates 1, Географічна інформація – Просторова прив’язка по координатах;
- ISO 19115:2003, Geographic Information — Metadata, Географічна інформація – Метадані;
- ISO 19118:2005, Geographic Information — Encoding, Географічна інформація – Кодування;
- ISO 19123:2005, Geographic Information — Schema for coverage geometry and functions, Географічна інформація – Схема для геометрії та функцій покриття;
- ISO/TS 19139: —1, Geographic Information — Metadata — XML schema implementation —1, Географічна інформація – Метадані – XML схема реалізації;
- ISO/IEC 19757-3: Information Technology – Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation — Schematron, Інформаційна технологія — Мови визначення схеми документів (DSDL) – Частина 3: Перевірка на основі правил – Schematron;
- ISO 80000-3, Quantities and units – Part 3: Space and time, Кількості та одиниці – Частина 3: Простір та час;
- ETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax. (August 1998), Єдині ідентифікатори ресурсів (URL): Загальний Синтаксис (серпень 1998 р.)
- W3C XLink, XML Linking Language (XLink) Version 1.0. W3C Recommendation (27 June 2001), Мова XML зв’язування (XLink) Версія 1.0 W3C рекомендація (27 червня 2001 р.);
- W3C XML, Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation (4 February 2004), Розширювана мова розмітки (XML) 1.0 (друге видання), W3C рекомендація (4 лютого 2004 р.);
- W3C XML Namespaces, Namespaces in XML. W3C Recommendation (14 January 1999), Імена пробілів у XML. W3C рекомендація (14 січня 1999 р.);
- W3C XML Schema Part 1, XML Schema Part 1: Structures. W3C Recommendation (2 May 2001), Частина 1, XML схема Частина 1: Структури. W3C рекомендація (2 травня 2001 р.);
- W3C XML Schema Part 2, XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001), Частина 2, XML схема частина 2: Типи даних. W3C рекомендація (2 травня 2001 р.)

## **4 ТЕРМІНИ ТА ПОЗНАЧЕННЯ**

### **4.1 Терміни та визначення**

Для цілей цього документу застосовані наступні терміни та визначення.

**4.1.1 прикладна схема** (*application schema*)

**Концептуальна схема** для даних, що потрібна для одного чи більше застосувань [ISO 19101:2002].

**4.1.2 асоціація** (*association*)

Семантичне відношення між двома чи більше класифікаторами, що визначає поєднання їх випадків [ISO 19501:2005].

**4.1.3 атрибут <XML>** (*attribute <XML>*)

Пара ім'я-значення, що міститься в елементі.

**Примітка 1.** У цьому документі атрибут є XML атрибутом, якщо він не визначений по-іншому. Синтаксис атрибуту XML є "Attribute::= Name = AttValue". Атрибут звичайно виступає як модифікатор елемента XML (наприклад, <Дорога gml:id = "d1" />; тут gml:id є атрибутом.

**4.1.4 межа** (*boundary*)

Множина, якою подають границі сутності [ISO 19107:2003].

**4.1.5 успадкований елемент <XML>** (*child element <XML>*)

Безпосередній елемент-нащадок іншого елемента.

**4.1.6** (*closure*)

Сполучення внутрішньої області та межі топологічного чи геометричного об'єкту [ISO 19107:2003].

**4.1.7 список кодів** (*codelist*)

Домен значень, що включає код для кожного припустимого значення.

**4.1.8 простір коду** (*codespace*)

Правило або право на код, ім'я, термін чи категорію.

*Приклад 1.* Приклади просторів коду включають словники, списки кодів тощо.

**4.1.9 композитна крива** (*composite curve*)

Послідовність кривих, у якій кожна крива (за виключенням першої) починається в кінцевій точці попередньої кривої в послідовності [ISO 19107:2003].

**Примітка 1.** Композитна крива, як множина прямих позицій, має всі властивості кривої.

**4.1.10 композитне тіло** (*composite solid*)

Зв'язана множина тіл, що межують одне з іншим через спільні поверхні [ISO 19107:2003].

**Примітка 1.** Композитне тіло, як множина прямих позицій, має всі властивості тіла.

**4.1.11 композитна поверхня** (*composite surface*)

Поєднана множина поверхонь, суміжних одна з іншою через спільні граничні лінії [ISO 19107:2003].

**Примітка 1.** Композитна поверхня як множина прямих позицій має всі властивості поверхні.

**4.1.12 координата** (*coordinate*)

Одне число з послідовності  $n$  чисел, що визначають позицію точки в  $n$ -мірному просторі [ISO 19111:2007].

**Примітка 1.** В референційній системі координат  $n$  чисел мають бути в певних одиницях.

**4.1.13 референційна система координат** (*coordinate reference system*)

Система координат, що пов'язана із об'єктом через дату [ISO 19111:2007].

**4.1.14 система координат** (*coordinate system*)

Набір математичних правил для призначення точкам координат [ISO 19111:2007].

**4.1.15 кортеж координат** (*coordinate tuple*)

Кортеж, що складається з послідовності координат [ISO 19111:2007].

**4.1.16 покриття** (*coverage*)

Об'єкт, що діє як функція, що повертає значення в межах діапазону для будь-якої прямої позиції в межах свого просторового, часового чи просторово-часового домену [ISO 19123:2005].

**4.1.17 крива** (*curve*)

Одновимірний геометричний примітив, що являє безперервне зображення [ISO 19107:2003].

**Примітка 1.** Межею кривої є множина точок на кінцях кривої. Якщо крива є колом, то два її кінця збігаються, і крива (якщо вона топологічно замкнена) вважається такою, що не має межі. Перша точка називається початковою, а остання точка – кінцевою точкою. Зв'язність кривої забезпечено вимогою “безперервного зображення лінії”. Топологічна теорема твердить, що безперервне зображення зв'язної множини є з'єднаним.

**4.1.18 тип даних** (*data type*)

Визначення домену значень та операцій, дозволених на значеннях цього домену [ISO/TS 19103:2005].

*Приклади:* ціле число, дійсне число, текст, булевський вираз, дата (конвертація даних в серії кодів).

**Примітка 1.** Типи даних включають примітивні попередньо визначені типи та типи, які може визначити користувач. Всі приклади типів даних не є ідентичними.

**4.1.19 дата** (*datum*)

Параметр або множина параметрів, що визначає позицію початку, шкалу та орієнтацію координатної системи [ISO 19111:2007].

**Примітка 1.** Дата може бути геодезичною датою, вертикальною датою, інженерною датою, датою зображення або часовою датою.

**4.1.20 пряма позиція** (*direct position*)

Позиція, описана одним набором координат у референційній системі координат [ISO 19107:2003].

**4.1.21 домен** (*domain*)

Чітко визначена множина [ISO/TS 19103:2005].

**Примітка 1.** На цій множині можна визначити математичну функцію, тобто в функції  $f:A \rightarrow B$   $A$  є доменом функції  $f$ .

**Примітка 2.** Домен пов'язаний із предметом або предметною областю.

**4.1.22 ребро** (*edge*)

Одновимірний топологічний примітив [ISO 19107:2003].

**4.1.23 елемент <XML>** (*element <XML>*)

Базова одиниця інформації в документі XML, що містить успадковані елементи, атрибути та текстові дані.

**Примітка 1.** З набору інформації XML: “Кожен XML документ містить один чи більше елементів, межі яких визначені початковими та кінцевими тегами, або, для порожніх елементів, тегами порожнього елемента. Кожен елемент характеризується типом, ідентифікований за іменем, що іноді називають загальним ідентифікатором (GI), та може мати набір специфікацій атрибутів. Кожна специфікація атрибуту має ім'я та значення”.

**4.1.24 оточення** (*exterior*)

Різниця між загалом та охопленою областю [ISO 19107:2003].

**4.1.25 грань** (*face*)

Двохвимірний топологічний примітив [ISO 19107:2003]

**Примітка 1.** Геометричною реалізацією грані є поверхня. Межа поверхні є множиною спрямованих ребер у тому ж топологічному комплексі, що асоційований із гранню через відношення ребер. Вони можуть бути організовані як кільця.

#### **4.1.26 географічний об'єкт** (*feature*)

Абстракція явища реального світу [ISO 19101:2002].

**Примітка 1.** Об'єкт може виступати як тип або екземпляр. Тип або екземпляр об'єкту треба використовувати, якщо тільки один із них мають на увазі.

#### **4.1.27 асоціація об'єктів** (*feature association*)

Відношення, що пов'язує екземпляри об'єктів одного типу з екземплярами такого ж або іншого типу об'єктів [ISO 19110:2005].

#### **4.1.28 функція** (*function*)

Правило, за яким кожен елемент з домену (джерела, або домену функції) асоційований із унікальним елементом з іншого домену (джерело, кодомен, або діапазон) [ISO 19107:2003].

#### **4.1.29 геодезична дата** (*geodetic datum*)

Дата, що описує відношення двох- або трьохвимірної координатної системи до Землі [ISO 19111:2007].

#### **4.1.30 геометричний об'єкт** (*geometric object*)

Просторовий об'єкт, що являє геометричну множину [ISO 19107:2003].

#### **4.1.31 геометричний примітив** (*geometric primitive*)

Геометричний об'єкт, що являє простий, зв'язаний та однорідний елемент простору [ISO 19107:2003].

#### **4.1.32 геометрична множина** (*geometric set*)

Множина прямих позицій [ISO 19107:2003].

#### **4.1.33 геометрична властивість <GML>** (*geometry property <GML>*)

Властивість GML географічного об'єкту, що описує деякий аспект геометрії об'єкту.

**Примітка 1.** Ім'я геометричної властивості є роллю геометрії відносно об'єкту.

#### **4.1.34 GML прикладна схема** (*GML application schema*)

Прикладна схема, написана в XML схемі згідно з правилами, визначеними в цьому міжнародному стандарті.

#### **4.1.35 GML документ** (*GML document*)

XML документ із кореневим елементом, що є одним із елементів AbstractFeature, Dictionary або TopoComplex, визначеним у GML схемі, або будь-який елемент групи заміни будь-яких із цих елементів.

#### **4.1.36 GML профіль** (*GML profile*)

Піднабір GML схеми.

#### **4.1.37 GML схема** (*GML schema*)

Компоненти схеми в XML просторі імен <http://www.opengis.net/gml/3.2>, як визначено в цьому міжнародному стандарті.

#### **4.1.38 регулярна мережа** (*grid*)

Сітка, що складається з двох чи більше множин ліній, у якій елементи кожної з множин перетинають елементи інших множин в алгоритмічний спосіб [ISO 19123:2005].

**Примітка 1.** Лінії поділяють простір на комірки.

#### **4.1.39 внутрішня область** (*interior*)

Множина всіх прямих позицій на геометричному об'єкті, але не на його межі [ISO 19107:2003].

#### **4.1.40 ламана лінія** (*line string*)

Лінія, що складається із прямолінійних відрізків.

#### **4.1.41 міра <GML>** (*measure <GML>*)

Значення, описане числовим обсягом із шкалою або із допомогою скалярної референцної системи.

**Примітка 1.** Коли використовують як іменник, міра є синонімом фізичної кількості.

#### **4.1.42 вимірювана величина** (*measurand*)

Певна кількість, яку треба вимірювати.

#### **4.1.43 простір імен <XML>** (*namespace <XML>*)

Колекція імен, ідентифікована URI посиланням, що можна використати в XML документах як імена елементів та атрибутів [W3C XML Namespaces].

#### **4.1.44 вузол** (*node*)

Нульвимірний топологічний примітив [ISO 19107:2003].

#### **4.1.45 об'єкт** (*object*)

Сутність із чітко визначеними межами, що ідентифікує та охоплює стан та поведінку [ISO 19107:2003].

**Примітка 1.** GML об'єкт є XML елементом типу, отриманого з AbstractGMLType.

#### **4.1.46 спостережний тип** (*observable type*)

Тип даних, що ідентифікує фізичну кількість як результат спостереження.

#### **4.1.47 точка** (*point*)

Нульвимірний геометричний примітив, що являє позицію [ISO 19107:2003]

**Примітка 1.** Межа точки є порожньою множиною.

#### **4.1.48 полігон** (*polygon*)

Плоска поверхня, визначена однією зовнішньою межею та нулем або більше внутрішніх меж

#### **4.1.49 властивість <GML>** (*property <GML>*)

Успадкований елемент GML об'єкту.

**Примітка 1.** Він відповідає атрибуту та ролі географічного об'єкту в асоціації в ISO 19109. Якщо властивість GML об'єкту має атрибут `xlink:href`, що посилається на об'єкт, то властивість виступає як роль об'єкту в асоціації.

#### **4.1.50 фізична величина** (*physical quantity*)

Величина, яку використовують для кількісного опису фізичного явища [ISO 31-0:1992].

**Примітка 1.** В GML фізична величина є завжди значенням, що описане з допомогою числа із шкалою або з допомогою скалярної референцної системи. Фізична кількість є синонімом міри, якщо останню використовують як іменник.

#### **4.1.51 діапазон** (*range*)

Множина всіх значень, що може приймати функція  $f$  як аргументи в межах її домену.

#### **4.1.52 прив'язана мережа** (*rectified grid*)

Мережа, для якої існує афінна трансформація між координатами мережі та зовнішньою референцною системою координат [ISO 19123:2005].

#### **4.1.53 схема** (*schema*)

Формальний опис моделі [ISO 19101:2002].

**Примітка 1.** Взагалі, схема є абстрактним поданням характеристик об'єкту та його відношень до інших об'єктів. У XML схемі подають відношення між атрибутами та елементами XML об'єкту (наприклад, документ та фрагмент документу)

#### **4.1.54 схема <XML Schema> (*schema <XML Schema>*)**

Колекція компонентів схеми в одному й тому ж цільовому просторі імен.

*Приклад 1.* Компонентами схеми W3C XML Schema є типи, елементи, атрибути, групи тощо.

#### **4.1.55 документ схеми <XML Schema> (*schema document <XML Schema>*)**

XML документ, що містить визначення компонентів схеми та декларацій.

**Примітка 1.** Схема W3C XML Schema забезпечує XML обмінний формат для інформації про схему. В одному документі схеми надають описи компонентів, асоційованих із певним XML простором імен, але кілька документів можуть описувати компоненти в тій же схемі, тобто в тому ж цільовому просторі імен.

#### **4.1.56 семантичний тип (*semantic type*)**

Категорія об'єктів, що поділяють деякі спільні характеристики і тому мають ідентифікаційне ім'я типу в певному домені або предметній області.

#### **4.1.57 послідовність (*sequence*)**

Конечна, впорядкована колекція пов'язаних елементів (об'єктів або значень), що можуть повторюватись [ISO 19107:2003].

#### **4.1.58 множина (*set*)**

Невпорядкована колекція пов'язаних елементів (об'єктів або значень), що не можуть повторюватись [ISO 19107:2003].

#### **4.1.59 просторовий об'єкт (*spatial object*)**

Об'єкт, використаний для подання просторової характеристики об'єкту [ISO 19107:2003].

#### **4.1.60 поверхня (*surface*)**

Двохвимірний геометричний примітив, яким локально подають безперервне зображення області чи площини.

**Примітка 1.** Межа поверхні є набором орієнтованих замкнених ліній, які окреслюють межі поверхні. Поверхні, ізоморфічні до сфери, або до  $n$ -torus (топологічна сфера з  $n$  "ручок") не мають межі. Такі поверхні називають циклами [ISO 19107:2003]

#### **4.1.61 тег <XML> (*tag <XML>*)**

Розмітка в документі XML, що визначає контент елемента.

*Приклад 1.* <Дорога>

**Примітка 1.** Тег без нахиленої наперед риски (наприклад, <Дорога> ) називають початковим тегом, а з наперед нахиленою рискою (тобто, </Дорога>) називають кінцевим тегом.

#### **4.1.62 топологічний об'єкт (*topological object*)**

Просторовий об'єкт, яким подають просторові характеристики, що зберігаються під час безперервних трансформацій [ISO 19107:2003].

#### **4.1.63 кортеж (*tuple*)**

Впорядкований список значень.

**Примітка 1.** Число значень у кортежі є незмінним.

#### **4.1.64 UML прикладна схема (*UML application schema*)**

Прикладна схема, написана на UML згідно з ISO 19109.

**4.1.65 уніформний ідентифікатор ресурсів (URI) (*Uniform Resource Identifier (URI)*)**

Унікальний ідентифікатор для ресурсу, структурований згідно до IETF RFC 2396.

**Примітка 1.** Загальний синтаксис є <scheme>::<scheme-specific-part>. Ієрархічний синтаксис в просторі імен є <scheme>://<authority><path>?<query>, див. [RFC 2396].

#### 4.2 Символи та скорочення

У цьому документі використані такі символи та скорочення:

- PCK референцна система координат;
- CRS референцна система координат;
- CSV Comma Separated Values – значення, відокремлені комою;
- EPSG European Petroleum Survey Group – Європейська група дослідження нафти;
- GIS Geographic Information System – географічна інформаційна система;
- GML Geography Markup Language – географічна мова розмітки;
- HTTP Hypertext Transfer Protocol – протокол передачі гіпертексту;
- IETF Internet Engineering Task Force – IETF;
- ISO International Organization for Standardization – Міжнародна організація стандартизації;
- OGC Open Geospatial Consortium – Відкритий геопросторовий консорціум;
- RDF Resource Description Framework – структура опису ресурсів;
- RFC Request for Comments – запит коментарів;
- SMIL Synchronized Multimedia Integration Language – мова синхронізованої мультимедійної інтеграції;
- SOAP Simple Object Access Protocol – протокол доступу до простих об'єктів;
- SVG Scalable Vector Graphics – масштабована векторна графіка;
- UML Unified Modeling Language – уніфікована мова моделювання;
- URI Uniform Resource Identifier – уніформний ідентифікатор ресурсу;
- URL Uniform Resource Locator – уніформний локатор ресурсу;
- URN Uniform Resource Name – уніформне ім'я ресурсу;
- W3C World Wide Web Consortium – Всесвітній веб-консорціум;
- WFS Web Feature Service – веб-сервіс об'єктів;
- XML eXtensible Markup Language – розширена мова розмітки;
- XSLT eXtensible Stylesheet Language – Transformations – розширена мова аркушів стилів – трансформації.

## 5 ПОЗНАКИ ТА СКОРОЧЕННЯ

### 5.1 XML простори імен

Всі компоненти схеми GML визначені в просторі імен з ідентифікатором "<http://www.opengis.net/gml/3.2>", для якого префікс gml або простір імен за умовчанням використовують у цьому міжнародному стандарті.

Всі компоненти, описані рекомендаціями W3C Xlink, визначають у просторі імен з ідентифікатором "<http://www.w3.org/1999/xlink>", для якого префікс xlink використовують у цьому міжнародному стандарті.

**Примітка 1.** Компоненти схеми в обох просторах імен задокументовані в документах XML схеми в додатку В.

## 5.2 Облік версій

Кожен документ схеми, що визначає компоненти GML схеми, має бути схарактеризований атрибутом версії, як це визначено в рекомендаціях XML Schema. Формат атрибуту версії є текстом x.y.z, де x означає головний номер версії, y означає допоміжний номер версії та z означає версію цього документу з виправленими помилками.

Версія, описана цим Міжнародним стандартом, є 3.2.1.

**Примітка 1.** Це є першою версією GML як міжнародного стандарту. Попередні версії були розроблені та опубліковані Відкритим геопросторовим консорціумом (OGC).

## 5.3 Скасовані частини попередніх версій GML

Дієслово "скасований" надає зауваження, що згаданий фрагмент цього Міжнародного стандарту збережений для сумісності з попередніми версіями, але може бути усунений з майбутньої версії без подальших посилань.

Розділи цього міжнародного стандарту, що описують скасовані GML компоненти або посилаються на них, виділені *курсивом*.

**Примітка 1.** Цей міжнародний стандарт опублікований Відкритим геопросторовим консорціумом як GML 3.2.1, а попередня ухвалена версія GML була зазначена Відкритим геопросторовим консорціумом як 3.1.1.

**Примітка 2.** Всі компоненти схеми, що входили до версії 2.1 GML, але були усунені з версії 3.0 GML, видалені та більше не підтримуються цим міжнародним стандартом.

## 5.4 UML Нотації

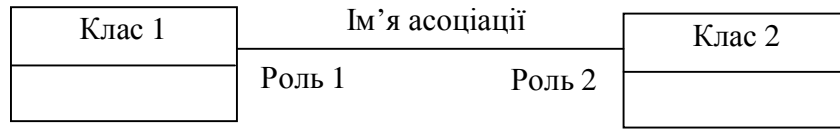
Низка діаграм, що зустрічаються в цьому міжнародному стандарті, подані з використанням діаграм статичної структури уніфікованої мови моделювання. UML нотації, використані в цьому міжнародному стандарті, описані на рисунку 1.

У цьому міжнародному стандарті використані такі стереотипи UML класів:

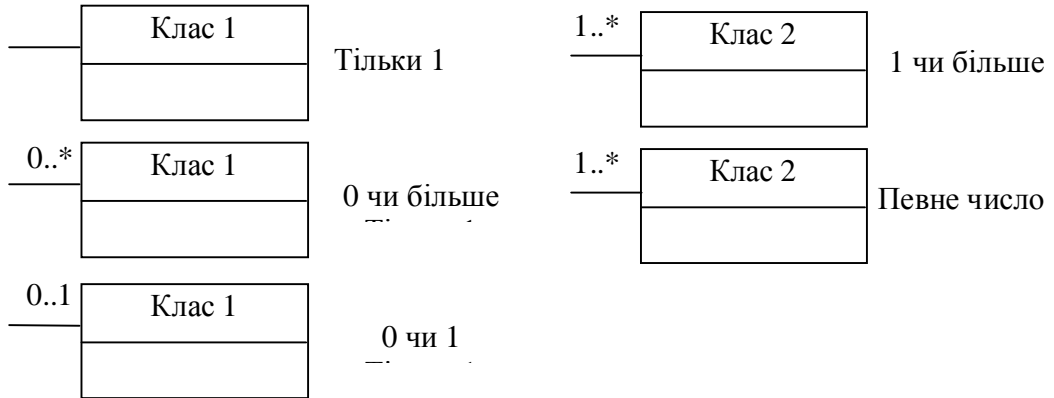
- <<DataType>>, що є набором властивостей, яким бракує ідентичності (незалежного існування та можливостей побічних ефектів); DataType (тип даних) є класом без операцій, першим призначенням якого є зберігання інформації;
- <<Union>> (об'єднання), що є набором властивостей; семантика є такою, що можна подати тільки одну з властивостей в певний час;
- <<FeatureType>> (тип географічного об'єкту), що є об'єктом, як визначено в ISO 19109;
- <<CodeList>> (список кодів), що є гнучким переліком, в якому використані значення в формі текстових рядків для подання списку потенційних значень;
- <<Enumeration>> (перелік), що є фіксованим списком дійсних ідентифікаторів поійменованих літерних значень; атрибути переліченого типу можуть мати значення тільки з цього списку;
- <<Abstract>>, що є абстрактним типом об'єкту (стереотип використовують у доповнення до форматування імені класу курсивом);
- <<Type>>, що є набором абстрактних атрибутів та асоціацій та означає, що їх специфікація не передбачає, що їх треба конкретно реалізовувати як змінні екземпляри;



Асоціації між класами



Валентність асоціацій



Агрегація між класами асоціацій



Спадкування класів (виділення підтипів)

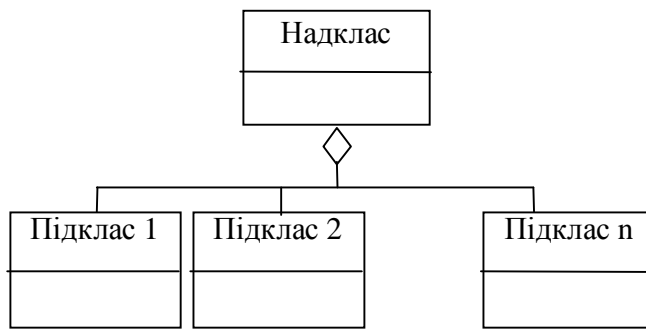


Рисунок 1 – UML нотація

У цьому стандарті використані такі типи даних:

- **CharacterString** – символний рядок, тобто послідовність символів (взагалі цей тип даних перекладають у “string” (текстовий рядок) у XML схемі);
- **Integer** – ціле число (взагалі цей тип даних перекладають в “integer”, тобто ціле число, у XML схемі);
- **Real** – число з плаваючою комою (взагалі цей тип даних перекладають як “double”, тобто довге дійсне число в XML схемі);
- **Boolean** – значення, що визначає істину чи хибність (взагалі цей тип даних перекладають у “boolean” в XML схемі).

### **5.5 XML схема**

У нормативних частинах цього міжнародного стандарту треба використовувати мову W3C XML схеми для опису граматики сумісних GML екземплярів даних. XML схема є багатою мовою з численними можливостями. Тоді як читач, незнайомий із XML схемою, може розуміти опис у загальних термінах, цей міжнародний стандарт не призначений служити в якості вступу до XML схеми. Для того, щоб мати повне розуміння цього міжнародного стандарту, читач повинен мати достатнє знання XML схеми.

## **6 ОГЛЯД GML СХЕМИ**

### **6.1 GML схема**

GML визначає XML кодування низки концептуальних класів, визначених в комплексі міжнародних стандартів ISO 19100 та абстрактній специфікації OpenGIS, згідно до цих стандартів та специфікацій.

Доречні концептуальні моделі включають такі, що визначені в:

- ISO/TS 19103 – Conceptual schema language (units of measure, basic types), Мова концептуальної схеми (одиниці виміру, базові типи);
- ISO 19107 – Spatial schema (spatial geometry and topology), Просторова схема (просторова геометрія та топологія);
- ISO 19108 – Temporal schema (temporal geometry and topology, temporal reference systems), Часова схема (часова геометрія та топологія, часові референційні системи);
- ISO 19109 – Rules for application schemas (features), Правила для прикладних схем (географічні об’єкти);
- ISO 19111 – Spatial referencing by coordinates (coordinate reference systems), Просторова прив’язка по координатах (референційні системи координат);
- ISO 19123 – Coverages, Покриття.

У багатьох випадках перехід від концептуальних класів до XML прямий, тоді як іноді він складніший. В будь-якому разі, такий перехід детально задокументований в додатку Г.

Крім того, GML забезпечує XML коди додатковими концепціями, ще не змодельованими в комплексі міжнародних стандартів ISO 19100 та абстрактної специфікації OpenGIS. Прикладами є рухомі об’єкти, спостереження та об’єкти значень. “Відсутні” концептуальні класи також визначені в додатку Г.

GML схема охоплює компоненти (XML елементи, атрибути, прості типи, комплексні типи, групи атрибутів, групи тощо), що описані в цьому Міжнародному стандарті. XML коди задовольняють вимоги стандарту ISO 19118.

### **6.2 GML прикладні схеми**

Розробники GML прикладних схем можуть розширювати або обмежувати типи, визначені в схемі GML, для визначення потрібних типів для домену аплікації. Вони можуть також напряду використовувати неабстрактні елементи, атрибути та типи з GML схеми у прикладній схемі, якщо зміни не потрібні.

Згідно до ISO 19109, типи об'єктів даної аплікації чи її домену визначені в прикладній схемі. GML прикладну схему треба визначити в XML схемі та імпортувати GML схему. Вона може бути побудована в один із двох різних способів:

- через дотримання правил для GML прикладних схем, наведених у пункті 21, де описані правила створення GML прикладної схеми прямо в XML схемі;
- через дотримання правил для прикладних схем в UML, відповідно до обмежень на такі схеми та правил їх перекладу в GML прикладні схеми, визначених у додатку Д цього міжнародного стандарту; перехід від прикладної схеми в UML, що узгоджена із ISO 19109, у відповідну GML прикладну схему базується на наборі правил кодування. Ці правила кодування відповідають правилам для GML прикладних схем та ISO 19118.

Другий підхід взагалі рекомендовано для належного використання структури концептуального моделювання комплексу міжнародних стандартів ISO 19100. Однак, наступні аргументи є прикладами того, коли перший підхід є виправданим:

- може виникнути потреба в додаткових можливостях GML схеми, крім тих можливостей, що доступні завдяки правилам кодування, визначеним у додатку Д;
- може бути потрібним тільки XML подання, і прикладна схема може бути порівняно простою, тому використання мови концептуальної схеми можна вважати за надмірне;
- аплікація може потребувати більш оптимізованого компактного XML кодування ніж те, що можна отримати в результаті застосування правил кодування, визначених у додатку Д.

**Примітка 1.** У додатку Е надано правила перекладу GML прикладної схеми в схему в UML, що задовольняє вимоги ISO 19109.

В обох випадках у GML схемах, узгоджених із цим міжнародним стандартом, мають використовувати всі застосовні компоненти GML схеми, як безпосередньо, так і за спеціалізацією, та такі GML прикладні схеми є дійсними згідно до правил для XML схеми. Спосіб створення GML прикладних схем не впливає на відповідність вимогам цього міжнародного стандарту.

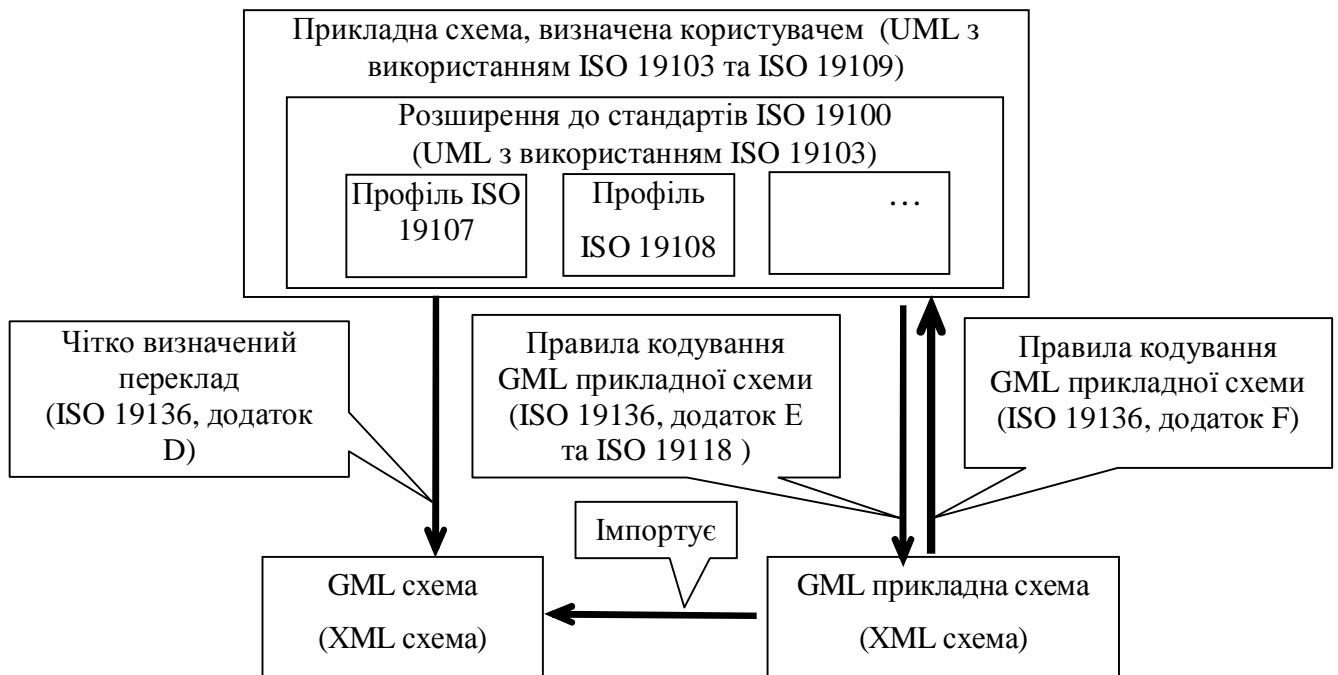
### **6.3 Відношення між комплексом міжнародних стандартів серії ISO 19100, GML схемою та GML прикладними схемами**

Підхід, прийнятий у цьому Міжнародному стандарті, показаний на рис. 2. Двома головними аспектами є:

- чітке документування концептуальної моделі GML: профіль комплексу Міжнародних стандартів ISO 19100, що реалізований з допомогою GML, задокументований, як і розширення цього профілю;

– підтримка розвитку прикладної схеми як у UML, так і в XML: для досягнення цього переходу між UML (тобто, від прикладних схем у UML, відповідних ISO 19109) та схемами XML (тобто, прикладних схем GML схемі XML ) засоби побудови, використані в обох поданнях, обмежені. В той час, коли це зменшує виключність описів схем до деякої міри, це також скорочує складність та допомагає впроваджувати їх.

**Примітка 1.** Тоді як переклад з UML до XML схем розглянутий в ISO 19118, Додаток А, зворотній перехід не розглянутий у жодному з міжнародний стандартів комплексу ISO 19100.



**Рисунок 2 -** Відношення між серією міжнародних стандартів ISO 19000 та ISO 19136/GML

#### 6.4 Перетворення компонентів GML схеми в документи схеми

GML визначає різні сутності, такі як об'єкти, геометрії, топології і т. і. через ієрархію GML типів об'єктів, як показано на UML діаграмі на рис. 3. Перехід між GML типами об'єктів та класами в концептуальній моделі комплексу міжнародних стандартів ISO 19100 та абстрактної специфікації OGC показаний в таблиці Г.2. Нормативна GML схема організована навколо цих типів об'єктів.

У підрозділі 7.2 описано базову схему GML компонентів. Вона визначає кореневий об'єкт, `gml:AbstractObject`, та корінь ієрархії GML класів, `gml:AbstractGML`.

У підрозділі 8.1 описано Xlink схему. Ця схема є OGC реалізацією Xlink специфікації з допомогою XML схеми. Її можуть у подальшому замінити еквівалентною схемою W3C.

**Примітка 1.** У цьому міжнародному стандарті опис XML схеми надано для xlink компонентів. Це здійснено для зручності в контексті середовища на базі XML схеми. Нормативні визначення викладені не в формі XML схеми в Xlink Рекомендації.

У підрозділі 8.2 визначено подання з деяких базових типів даних допомогою GML, які використовують у GML схемі. Більшість цих типів є простими типами або простими типами контенту.

У розділі 9 описано компоненти схеми об'єкту, які визначають `gml:AbstractFeature` та деякі похідні компоненти.

У розділах 10 і 11 та пункті 10.5.10 описано компоненти схеми геометрії, що визначають `gml:AbstractGeometry`, `gml:AbstractGeometricPrimitive`, `gml:AbstractGeometricAggregate`, `gml:GeometricComplex` та деякі інші похідні компоненти.

У розділі 12 описано компоненти схеми референційної системи координат, що визначають підтипи `gml:IdentifiedObject`, `gml:AbstractCRS`, `gml:AbstractCoordinateReferenceSystem`, та елементи і типи, потрібні для побудови специфічних референційних систем координат.

У розділі 13 описано компоненти схеми для топології, які визначають `gml:AbstractTopology`, `gml:AbstractTopoPrimitive`, `gml:TopoComplex` та деякі інші похідні компоненти.

У розділі 14 описано компоненти схеми для часових конструктивів, що визначають `gml:AbstractTimeObject`, `gml:AbstractTimePrimitive`, `gml:AbstractTimeGeometricPrimitive`, `gml:AbstractTimeTopologyPrimitive`, `gml:AbstractTimeComplex` та похідні компоненти, як і `gml:DynamicFeature` та похідні компоненти.

У розділі 15 описано компоненти схеми для визначень та словників, включно із `gml:Definition` та `gml:Dictionary`.

У розділі 16 описано компоненти схеми для побудови одиниць вимірювання (`gml:UnitDefinition` та похідні компоненти), мір та об'єктів значень (`gml:AbstractValue`, `gml:AbstractScalarValue`, `gml:AbstractScalarValueList` та похідні компоненти).

У розділі 17 описано схему компонентів для описування напрямку.

У розділі 18 описано компоненти схеми для спостережень (`gml:Observation` та похідні компоненти).

У розділі 19 описано компоненти схеми для мереж та покриттів `gml:Grid`, `gml:AbstractCoverage`, `gml:AbstractDiscreteCoverage`, `gml:AbstractContinuousCoverage` та похідні компоненти.

У цих підрозділах описано нормативну GML схему та пояснено її зміст, структуру та залежності.

Подання GML схеми, описане в цьому стандарті, здійснюється на мові обмінного формату XML, забезпеченого W3C XML схемою. Описи набору компонентів розміщено в документи схеми, де в кожному документі зібрано компоненти, що широко відповідає класифікації, показаній на рис. 2. Однак, тоді як подання на мові XML кожного компоненту GML схеми у цьому стандарті є нормативним, розподіл по пакетах у документи схеми не є таким. У розділі 20 (профілі) та додатку Ж (створення підмножин) описано принципи та методи для альтернативного визначення пакетів в XML поданні компонентів GML схеми.

Всі компоненти визначені або задекларовані в цьому Міжнародному стандарті з використанням того ж самого цільового простору імен з <http://www.opengis.net/3.2>.

**Примітка 2.** XML простір імен забезпечує механізм для запобігання неоднозначності через конфлікти імен в XML документах. Всі компоненти, описані в єдиному документі схеми,

перебувають у єдиному цільовому просторі імен, але більш ніж один документ схеми може описувати компоненти в просторі імен. У колах розробників XML є прецеденти для призначення одного чи кількох просторів імен для набору компонентів схеми для однієї й тієї ж аплікації. Використання єдиного простору імен для компонентів GML схеми є сумісним з необов'язковим перекладом подання компонентів GML в XML з одного документу схеми до іншого.

В UML використовують пакети для збирання пов'язаних компонентів. Більше того, в комплексі міжнародних стандартів ISO 19100, префікси, що відповідають зразку "AA\_", використовують для розрізнення між класами з різних пакетів, так, щоб вони нагадували XML простори імен. Однак, через вищезгадані причини, переклад компонентів GML не є нормативним, та всі компоненти GML перебувають у єдиному просторі імен, тобто не може бути відповідності між двома літерними префіксами з комплексу міжнародних стандартів ISO 19100 та XML просторами імен у GML.

### **6.5 Скасовані та експериментальні компоненти схеми**

Експериментальні, довідкові компоненти схеми, що стосуються правил для призначення стилів GML об'єктам за умовчанням, описані в додатку І.

Скасовані глобальні компоненти схеми (елементи, атрибути, типи) містяться в додатку К.

## **7 GML СХЕМА – ЗАГАЛЬНІ ПРАВИЛА ТА ОСНОВНІ КОМПОНЕНТИ СХЕМИ**

### **7.1 МОДЕЛЬ ТА СИНТАКСИС GML**

#### **7.1.1 Документи GML екземплярів**

У GML використовують явний синтаксис для конкретизації GML прикладної схеми, сумісний із загальною моделлю об'єктів, визначеною в ISO 19109 в XML документі.

Географічний об'єкт закодований як XML елемент з іменем типу географічного об'єкту. Інші об'єкти, які можна ідентифікувати, закодовані як XML елементи з іменем типу об'єкту.

Кожен атрибут об'єкту та роль об'єкту в асоціації є властивістю об'єкту. Властивості об'єкту закодовано в XML елементі.

**Примітка 1.** Термін "атрибут" в XML стосується певного синтаксичного компоненту в XML документах, і щоб уникнути непорозумінь при описі кодів XML, GML відповідає термінології RDF (W3C, 1999), і натомість використовують радше термін "властивість", ніж "атрибут" чи "роль в асоціації". У загальній моделі об'єктів (ISO 19109) також користуються терміном "властивість" як узагальненням понять "атрибут", "роль в асоціації" та "операція".

Більше того, семантика властивостей, яка виражена через ім'я елемента, через який подана властивість, відрізняється від значення властивості, що подане через контент елемента властивостей. Елемент властивості може містити її значення як включений закодований контент або посилатися на її значення через простий XLink. Значення властивості може бути простим або може бути географічним чи іншим складним об'єктом. Якщо значення простої властивості включене, то його записують як літерне значення без будь-якої вбудованої розмітки (текст), а якщо значення складне, воно є піддеревом із використанням XML розмітки (тобто, XML елементом з підструктурою).

**Примітка 2.** Модель у GML має явне подання з використанням профілю UML, використаного в комплексі міжнародних стандартах ISO 19100 (визначений у ISO/TS 19103). Це детально описане в додатках Г та Д, але може бути підсумовано приблизно і взагалі наступним чином.

Географічні об'єкти подані:

- в UML через об'єкти, де ім'я типу географічного об'єкту використане як і ім'я класу об'єктів;
- в GML екземплярах через XML елементи, де ім'я типу географічного об'єкту використане як ім'я елемента.

Властивості географічних об'єктів подані:

- у UML через ролі в асоціаціях з іншими класами типів географічних об'єктів і атрибути класів типів географічних об'єктів, де семантику властивостей подано іменем ролі в асоціації та іменем атрибуту;
- у GML екземплярах через піделементи (відомі як елементи властивостей), де семантику властивостей подано іменем елемента властивості.

Значення властивості має тип, зазначений:

- у UML через клас цілі асоціації або типу даних атрибуту;
- у GML, у випадку властивостей із комплексними значеннями, через ім'я елемента об'єкту, що міститься в елементі властивості та у випадку властивостей із простими значеннями – через літерне значення без будь-якої вбудованої розмітки XML.

Результатом є багатошаровий документ XML, у якому XML елементи, що відповідають географічним об'єктам, об'єктам чи значенням виявляються переплетеними з XML елементами, що відповідають властивостям, які пов'язують їх. Функція географічного об'єкту, об'єкту або значення може завжди бути визначена через перевірку імені елемента властивості, що напряму містить її або має посилання на неї.

**Примітка 3.** Цей підхід до кодування іноді згадують як модель “об'єкт-властивість”; він є основою моделі GML кодування, з часу ухвалення першої версії в OGC. Тоді як іноді цей підхід до кодування додає більше рівнів елементів у документи екземплярів, він забезпечує суттєві переваги: допомагає зробити документ екземпляру в GML зрозумілим, надає передбачувану структуру та забезпечує уникнення занадто довірливого покладання на схему, оскільки очікують, що GML документ екземпляру переживе широке використання схеми XML.

### 7.1.2 Лексичні домовленості

Є кілька лексичних позначень, використаних у GML схемі для імен елементів та комплексних типів, щоб сприяти людському сприйняттю GML екземплярів та схем:

- об'єкти наведені як XML елементи з концептуально змістовним ім'ям у верхньому реєстрі UpperCamelCase;
- властивості наведені як XML елементи, чиє ім'я знаходиться в нижньому реєстрі lowerCamelCase;
- абстрактні елементи мають префікс “Abstract” (об'єкти) або “abstract” (властивості), приєднаний спереду до їх імені;
- імена в комплексних типів у XML схемі знаходяться у верхньому реєстрі UpperCamelCase та закінчуються словом “Type”;
- комплексні типи абстрактної XML схеми мають попереду слово “Abstract”.

Наполегливо рекомендовано дотримуватись цих позначень також і в GML прикладних схемах. Правила можна застосувати тільки для мов, що розрізняють між великими та малими літерами.

**Примітка 1.** UpperCamelCase є позначкою, за якою ім'я утворено з кількох слів, що з'єднані разом як одне слово з першою великою літерою кожного слова всередині нового слова, що складає ім'я. lowerCamelCase є різновидом, у якому перша літера нового слова є малою, що допомагає швидко відрізнити його від імені в UpperCamelCase.

### 7.1.3 Визначення мови GML у XML схемі

GML схема складається з компонентів W3C XML схеми, що визначають типи та зазначають:

- XML елементи для кодування об'єктів GML з ідентифікацією;
- XML елементи для кодування у GML властивості цих об'єктів;
- XML атрибути, що кваліфікують ці властивості.

GML об'єкт є XML елементом типу, отриманого прямо чи непрямо з AbstractGMLType. З цього походження GML об'єкт повинен мати атрибут gml:id.

Властивість у GML не повинна походити від 12 мати атрибут gml:id або будь-який інший атрибут XML типу ID.

Елемент є у GML властивістю тоді і лише тоді, коли він є успадкованим елементом GML об'єкту.

GML об'єкт не може виступати безпосереднім нащадком GML об'єкту.

Тому жоден елемент не може одночасно бути GML об'єктом та GML властивістю.

Всі XML атрибути, задекларовані в GML схемі визначені без простору імен, за єдиним виключенням – XML атрибутом gml:id.

**Примітка 1.** Використання додаткових XML атрибутів у GML прикладній схемі не рекомендовано.

## 7.2 Компоненти схеми gmlBase

### 7.2.1 Цілі компонентів базової схеми

Компоненти схеми gmlBase установлюють модель та синтаксис GML, зокрема:

- кореневий XML тип, з якого мають бути отримані XML типи для всіх GML об'єктів;
- шаблони та компоненти для GML властивостей;
- шаблони для колекцій та масивів, та компоненти для загальних колекцій та масивів;
- компоненти для асоціювання метаданих з GML об'єктами;
- компоненти для побудови визначень та словників.

**Примітка 1.** Відповідний документ схеми в додатку В визначений таким незалежним від розміщення іменем (з використанням синтаксису URN):  
urn:ogc:specification:gml:schema-xsd:gmlBase:3.2.1

### 7.2.2 Базові об'єкти

#### 7.2.2.1 AbstractObject

Абстрактний елемент зручності gml:AbstractObject задекларований наступним чином:

```
<element name="AbstractObject" abstract="true"/>
```



Цей елемент немає визначеного типу, і тому є неявно зазначений (згідно до правил W3C XML схеми) як anyType XML схеми. Його використовують як голову групи заміни XML схеми, що уніфікує елементи складного контенту та певні елементи простого контенту для типів даних у GML, включно з групою заміни gml:AbstractGML.

**Примітка 1.** gml:AbstractObject визначений передусім, щоб він виступав як змінна в певних шаблонах агрегатів, де необхідно дозволити або елементам в групі заміни gml:AbstractGML, або певним елементам складного або простого контенту бути дійсними в екземплярі.

GML набір даних (також його називають екземпляром даних або документом даних) подають як GML об'єкт. Цей об'єкт може в свою чергу бути колекцією GML об'єктів.

### 7.2.2.2 AbstractGML, AbstractGMLType

Найголовніші компоненти для подання об'єктів, які можна ідентифікувати, описані в схемі наступним чином:

```
<element name="AbstractGML" type="gml:AbstractGMLType" abstract="true"
substitutionGroup="gml:AbstractObject"/>
<complexType name="AbstractGMLType" abstract="true">
  <sequence>
    <group ref="gml:StandardObjectProperties"/>
  </sequence>
  <attribute ref="gml:id" use="required"/>
</complexType>
<group name="StandardObjectProperties">
  <sequence>
    <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="gml:description" minOccurs="0"/>
    <element ref="gml:descriptionReference" minOccurs="0"/>
    <element ref="gml:identifier" minOccurs="0"/>
    <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</group>
```

Абстрактний елемент gml:AbstractGML є “будь-яким GML об'єктом, що має ідентичність”. Він діє як голова групи заміни XML схеми, яка може включати будь-який елемент, що є GML географічним об'єктом, або інший об'єкт, з ідентичністю. Його використовують як змінну в моделях контенту в GML ядрі та в прикладних схемах. Він є абстрактним надкласом для всіх GML об'єктів.

Складання пар із gml:AbstractGML та gml:AbstractGMLType показує базовий шаблон, використаний у GML схемі, з допомогою якої кожний GML об'єкт поданий через глобальну декларацію елемента, що має визначення асоційованого типу в XML схемі. Ім'я елемента, через який поданий GML об'єкт, відображає концептуальне значення об'єкту. Ім'я загальних елементів у GML включає gml:AbstractObject, gml:AbstractGML, gml:AbstractFeature, gml:AbstractValue, gml:AbstractCoverage, gml:AbstractTopology та gml:AbstractCRS. Ці загальні елементи, через які подають об'єкти, визначені і в інших місцях цього Міжнародного стандарту.

Успадковані XML елементи та XML атрибути GML об'єкту є властивостями цього об'єкту. Тому об'єкт, поданий через елемент gml:AbstractGML, має п'ять не

скасованих властивостей: `gml:identifier`, `gml:description`, `gml:descriptionReference`, `gml:name` та `gml:id`. Вони описані в пункті 7.2.4.

**Примітка 1.** Група `gml:StandardObjectProperties` надана для зручності побудови прикладної схеми, особливо коли треба визначити типи, отримані через обмеження з `gml:AbstractGMLType` та `gml:AbstractFeatureType`. Походження через обмеження вимагає, щоб усі компоненти, використані незмінними, були скопійовані в нове визначення типу. Як альтернатива включенню декларації елемента для всіх стандартних властивостей об'єкту, натомість можна використовувати посилання в один рядок на `gml:StandardObjectProperties`:

```
<group ref="gml:StandardObjectProperties"/>
```

## 7.2.3 GML властивості

### 7.2.3.1 Вступ

Термін “властивість” використовують у смислі GML властивості, що є будь-якою характеристикою GML об'єкту. Елемент у GML документі або у потоці даних є GML властивістю тоді і тільки тоді, коли він є успадкованим елементом GML об'єкту. Смысл кожної властивості має бути зазначений іменем елемента, що конкретизує її.

GML об'єкти можуть мати необмежене число властивостей у додаток до тих, що успадковані з `gml:AbstractGMLType`. Властивість можна визначити так, щоб вона мала простий або складний контент. Властивість із простим контентом має простий тип контенту XML схеми, що показано у випадку стандартних елементів властивості `gml:description` і `gml:name`. Властивість із складним контентом має складний тип контенту XML схеми.

Елементи властивості можуть використовуватись у двох режимах:

- включені значення властивості подають безпосередньо як контент елемента властивості; цей метод використовують у стандартній властивості `gml:name` та для `gml:description` (див. підпункт 7.2.4.2);
- через посилання: значення властивості наявне в іншому місці, і вона ідентифікована через атрибут `xlink:href` на елементі властивості; цей альтернативний метод можна застосувати для стандартної властивості `gml:descriptionReference` (див. підпункт 7.2.4.3).

*Приклад 1.* Приклади використання `xlink` посилань наведені в підрозділі 8.1

**Примітка 1.** Ролі об'єктів в асоціаціях, визначені в ISO 19109 (загальна модель об'єктів) та абстрактній специфікації OpenGIS, тема 8, можна подавати різними способами в GML прикладній схемі:

- через реалізацію тільки однієї ролі в асоціації як такої, по якій можна переміщатись, тобто, її подання в XML кодї; це є звичайним поданням у GML схемі як такої з деякими виключеннями, наприклад, ролей в асоціаціях меж та спільних меж між топологічними об'єктами;
- через визначення індивідуальних властивостей в типах об'єктів, що беруть участь у асоціації; однак, у цьому випадку, обмеження узгодженості, закладені в асоціації, не можуть бути гарантовані перевіркою XML схеми; цей стиль кодування, наприклад, використаний для ролей в асоціаціях меж топологічними об'єктами та в додатку Д; дивіться також підпункт 7.2.3.9;
- через створення об'єкту асоціації як GML об'єкту; це також дозволяє моделювати *n*-кратні асоціації та асоціації з властивостями;
- через використання розширених розширених посилань Xlinks; це кодування подібне до подання “об'єкту асоціації”.

### 7.2.3.2 AssociationAttributeGroup

Компоненти XLink є стандартним методом підтримки гіпертекстових посилань в XML схемі. Атрибутивна група `gml:AssociationAttributeGroup` надана для підтримки використання посилань Xlinks як метод для зазначення значення властивості через посилання в одноманітний спосіб в GML. Ця група атрибутів визначена наступними чином:

```
<attributeGroup name="AssociationAttributeGroup">
  <attributeGroup ref="xlink:simpleLink"/>
  <attribute name="nilReason" type="gml:nilReasonType"/>
  <attribute ref="gml:remoteSchema"/>
</attributeGroup>
```

з такими визначеннями з Xlink (дивіться підрозділ 8.1):

```
<attributeGroup name="simpleLink">
  <attribute name="type" type="string" fixed="simple" form="qualified"/>
  <attribute ref="xlink:href"/>
  <attribute ref="xlink:role"/>
  <attribute ref="xlink:arcrole"/>
  <attribute ref="xlink:title"/>
  <attribute ref="xlink:show"/>
  <attribute ref="xlink:actuate"/>
</attributeGroup>
```

Значення GML властивості, яка має атрибут `xlink:href`, є ресурсом, що повертається при проходженні по посиланню.

Атрибут `nilReason` можна використовувати в елементі властивості для зазначення підстави для нульового значення.

**Примітка 1.** Всі компоненти в атрибуті не є обов'язковими.

### 7.2.3.3 abstractAssociationRole, тип AssociationRole

Для підтримки кодування властивостей, що мають комплексний контент, базовий шаблон для елементів властивості, поданий у GML схемі, визначений таким чином:

```
<element name="abstractAssociationRole" type="gml:AssociationRoleType" abstract="true"/>
<complexType name="AssociationRoleType">
  <sequence minOccurs="0">
    <any namespace="##any"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Застосування цього шаблону обмежить чисельність об'єктів в елементі властивості за рахунок використання тільки однієї такої моделі контенту. Екземпляр цього типу має містити елемент, яким подано об'єкт, або служити як покажчик на віддалений об'єкт.

Застосування цього шаблону для визначення специфічного типу властивості прикладної схеми дозволяє обмежити:

- включений об'єкт – до специфічних типів об'єкту;

- кодування – до “тільки через посилання” (див. підпункт 7.2.3.7),
- кодування – до “тільки включене” (див. підпункт 7.2.3.8).

**Примітка 1.** Декларація `gml:abstractAssociationRole` та визначення її супроводжувального типу надане для зручності, тобто для того, щоб вона діяла як шаблон чи зразок для побудови елементів властивості в прикладних схемах. Немає вимоги до специфічних властивостей для використання отримання типу XML схеми від `gml:AssociationType` задля створення властивостей у відповідній GML прикладній схемі. Це контрастує з вимогами до отримання моделі контенту для всіх об’єктів, які можна ідентифікувати, з `gml:AbstractGMLType`, та для всіх географічних об’єктів з `gml:AbstractFeatureType`.

**Примітка 2.** У той час як `gml:abstractAssociationRole` є абстрактною, її тип `gml:AssociationRoleType` не є таким, тому що такий же тип використовують у властивості `gml:member`, яка може мати екземпляри (див. підпункт 7.2.3.10). Слід зауважити, що ця властивість скасована.

#### 7.2.3.4 Включати чи через посилання?

Елемент `any` у моделі контенту для властивостей є необов’язковим. У сполученні із потужностями компонентів в `AssociationAttributeGroup` це значить, що елемент цього типу може мати елемент контенту або атрибуту `Xlink` посилань. GML елементи властивості, в яких дотриманий цей шаблон, можна використовувати для приєднання властивостей через включення або через посилання.

*Приклад 1.* Властивість, надана для географічних об’єктів – “centerOf”. Її можна використати для зазначення просторового розміщення шляхом включення таким чином:

```
<gml:centerOf>
  <gml:Point gml:id="point96" srsName="urn:ogc:def:crs:EPSG::4326">
    <gml:pos>-31.936 15.834</gml:pos>
  </gml:Point>
</gml:centerOf>
```

де використаний об’єкт `gml:Point`, як визначено в GML схемах геометрії (описано в підрозділ 10.2). Той же елемент властивості можна використати для зазначення місцеположення через посилання таким чином:

```
<gml:centerOf xlink:href="http://my.big.org/locations/point53"/>
```

де “<http://my.big.org/location/point53>” ідентифікує точку (елемент `gml:Point`), яку постачає зазначена служба.

Однак, елемент властивості, що відповідає цьому шаблону, може не мати контенту або атрибутів, або може мати і контент, і атрибуту, і при цьому залишатись дійсним у XML схемі. Неможливо обмежити одночасну наявність контенту та атрибутів, тому неможливо використовувати W3C XML схему для обмеження властивості бути поданою або тільки через включення, або тільки через посилання.

Якщо і посилання, і контент є в екземплярі елементу властивості, то об’єкт, який можна знайти при проходженні по посиланню `xlink:href`, має бути нормативним значенням властивості. Об’єкт, включений як контент, може бути використаний отримувачем даних тільки якщо віддалений екземпляр не можна вирішити; це можна вважати “кешованою” версією об’єкту.

**Примітка 1.** Більшість оцінюваних властивостей GML об’єктів у GML схемі об’єктів можна закодувати шляхом включення або через посилання. Однак, використовуючи профіль

GML (див. розділ 20), можна обмежити використання “тільки включене” або “тільки через посилання”.

### 7.2.3.5 Власність на значення властивостей

Кодування GML властивості шляхом включення порівняно з кодуванням через посилання нічого не виражає стосовно “власності” GML об’єкту, що міститься або на який спрямоване посилання, тобто стиль кодування не повинен передбачати ніякого семантики “глибокого копіювання” або “глибокого видалення”. Щоб зазначити власність на GML об’єкт, що міститься або на який спрямоване посилання, можна додати атрибутивну групу `gml:OwnershipAttributeGroup` до елементів оціненої властивості об’єкту. Якщо атрибутивна група не є частиною моделі контенту такого елементу властивості, то значення не може бути “у власності”.

Атрибутивна група визначена наступним чином:

```
<attributeGroup name="OwnershipAttributeGroup">
  <attribute name="owns" type="boolean" default="false"/>
</attributeGroup>
```

Коли значенням атрибуту `owns` є "істина", то існування об’єктів, що включені або на які спрямоване посилання залежить від існування батьківського об’єкту.

*Приклад 1.* Якщо властивість “hasOwner” подана в документі екземпляру як

```
<Parcel gml:id="p123">
  <hasOwner xlink:href="urn:x-abc:id:o123"/>
</Parcel>
```

то об’єкт, на який спрямоване посилання, тобто особа, не є у власності (“owned”) об’єкту “земельна ділянка” (“parcel”), тобто об’єкт “особа” (“person”) не може бути вилучено, якщо вилучається земельна ділянка. Однак, якщо властивість закодована із атрибутом `owns="true"`, наприклад,

```
Car gml:id="c123">
  <hasParts owns="true" xlink:href="urn:x-abc:id:x123"/>
  <!-- ... -->
</Car>
```

то об’єкт, на який спрямоване посилання, є у власності (“owned”) об’єкту “авто” (“feature”), тобто, якщо видалити частину, то буде видалене і авто.

### 7.2.3.6 abstractStrictAssociationRole

Обмеження, які вимагають, щоб значення властивості було подано або через включення, або через посилання, можна точно описувати з допомогою допоміжної мови обмежень Schematron (див. ISO/IEC 19757-3). Абстрактні, глобальні елементи `gml:abstractAssociationRole` та `gml:abstractStrictAssociationRole` використовують `gml:AssociationRoleType`, але наступні фрагменти схеми показують, як декларація елементу може супроводжуватись обмеженнями Schematron, щоб обмежити властивість до поведінки або як включеної, або в режимі через посилання, але не в обома способами одночасно.

```
<element name="abstractAssociationRole" type="gml:AssociationRoleType" abstract="true"/>
<element name="abstractStrictAssociationRole" type="gml:AssociationRoleType" abstract="true"/>
<sch:schema
  xmlns:sch="http://purl.oclc.org/dsdl/schematron"
  xmlns:gml="http://www.opengis.net/gml/3.2">
```

```

xmlns:xlink="http://www.w3.org/1999/xlink" xml:lang="en">
  <sch:title>Schematron constraints for GML / ISO 19136</sch:title>
  <sch:ns prefix="sch" uri="http://purl.oclc.org/dsdl/schematron"/>
  <sch:ns prefix="gml" uri="http://www.opengis.net/gml/3.2"/>
  <sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
  <sch:pattern>
    <sch:rule context="gml:abstractStrictAssociationRole">
      <sch:assert test="not(@xlink:href and (*|text()))">Елемент властивості не
може мати і посилання на об'єкт, і містити об'єкт.</sch:assert>
      <sch:assert test="@xlink:href | (*|text())">Елемент повинен мати або
посилання на об'єкт, або містити об'єкт.</sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>

```

**Примітка 1.** В деяких XML валідатори запускають обмеження Schematron автоматично. Інакше, код Schematron можна розглядати як чисто формальний опис потрібного обмеження. Він тут включений головне, щоб показати, як його можуть використати розробники схем для спеціальних цілей.

### 7.2.3.7 abstractReference, ReferenceType

Для підтримки кодування властивостей, чії значення подають віддалено через посилання, надано такі компоненти:

```

<element name="abstractReference" type="gml:ReferenceType" abstract="true"/>
<complexType name="ReferenceType">
  <sequence/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

Елемент gml:abstractReference є абстрактним, і тому може бути використаний як голова групи заміни більш специфічних елементів, що забезпечують значення через посилання.

**Примітка 1.** Якщо gml:abstractReference є абстрактним, його тип gml:ReferenceType не є таким, тому що тип треба напряму використовувати у прикладній схемі, якщо в елементі властивості треба використовувати кодування “тільки через посилання”.

### 7.2.3.8 abstractInlineProperty, InlinePropertyType

рамки кодування властивостей, значення яких подані включеними, мають бути надані такі компоненти:

```

<element name="abstractInlineProperty" type="gml:InlinePropertyType" abstract="true"/>
<complexType name="InlinePropertyType">
  <sequence>
    <any namespace="##any"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Елемент `gml:abstractInlineProperty` є абстрактним, і тому може бути використаний як голова групи заміни більш специфічних елементів, що забезпечують значення як включене.

### 7.2.3.9 Властивості, що відображають те ж саме відношення

Якщо значення властивості об'єкту є іншим об'єктом, і цей об'єкт також містить властивість для асоціації між двома об'єктами, то це ім'я зворотної властивості можна закодувати в елементі `gml:reversePropertyName` в `appinfo` анотації властивості елементу, щоб задокументувати обмеження між двома властивостями. Значення елементу має містити кваліфіковане ім'я елементу властивості.

```
<element name="reversePropertyName" type="string"/>
```

*Приклад 1.*

```
<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:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

### 7.2.3.10 Властивості об'єктів значень

Об'єкти значень (див. підрозділ 16.4) є спеціальними об'єктами у тому розумінні, що окрема властивість, подана простим літерним значенням, виступає як прямий контент елементу об'єкту без додаткового елементу для властивості.

*Приклад 1.* `<gml:Integer>5</gml:Integer>` використовують замість, наприклад, `<gml:Integer><gml:value>5</gml:value></gml:Integer>`.

## 7.2.4 Стандартні властивості GML об'єктів

### 7.2.4.1 Отримання з типу AbstractGML

Типи XML схеми для всіх GML об'єктів можна отримати прямо чи непрямо з `gml:AbstractGMLType`. Це значить, що всі GML об'єкти успадковують певні стандартні властивості, що включені в модель контенту типу `gml:AbstractGMLType`.

#### 7.2.4.2 description

Значенням цієї властивості є текстовий опис об'єкту. У `gml:description` використовують `gml:StringOrRefType` та його модель контенту, то він може містити простий контент у формі текстового рядку.

```
<element name="description" type="gml:StringOrRefType"/>
```

**Примітка 1.** Використання опису `gml:description` для посилань на зовнішній опис скасовано та замінено на властивість `gml:descriptionReference` (див. підпункт 7.2.4.3).

### 7.2.4.3 descriptionReference

Значення цієї властивості є віддаленим текстовим описом об'єкту. Атрибут `xlink:href` властивості `gml:descriptionReference` має посилання на зовнішній опис

```
<element name="descriptionReference" type="gml:ReferenceType"/>
```

### 7.2.4.4 name, identifier

Властивість `gml:name` забезпечує напис або ідентифікатор об'єкту, звичайно описове ім'я.

Об'єкт може мати кілька імен, звичайно призначені різними організаціями. У `gml:name` використана модель контенту `gml:CodeType`. Власник імені зазначений значенням (необов'язковим) атрибуту `codeSpace`. Ім'я може бути або може не бути унікальним, як визначено правилами організації, відповідальної за `codeSpace`. У загальному користування є одне ім'я на організацію, тому програма може вибрати ім'я із того `codeSpace`, якому вона надасть перевагу.

```
<element name="name" type="gml:CodeType"/>
```

Часто в організації, що обслуговує, об'єкту призначають спеціальний ідентифікатор з тим, щоб його використовували в посиланнях на цей об'єкт. У цих випадках треба надати `codeSpace`. Цей ідентифікатор звичайно унікальний або глобальний у прикладному домені. `gml:identifier` є наперед визначеною властивістю для таких ідентифікаторів.

*Приклад 1.* UUID and URN є широко використовуваними ідентифікаторами.

```
<element name="identifier" type="gml:CodeWithAuthorityType"/>
```

### 7.2.4.5 id

Атрибут `gml:id` підтримує роботу з XML елементом, яким подають GML об'єкт. Його використання обов'язкове для всіх GML об'єктів.

```
<attribute name="id" type="ID"/>
```

Він є XML типом ID, тому він має бути унікальним в XML документі, у якому він зустрічається. Зовнішній ідентифікатор для XML елемента, яким подають GML об'єкт у формі URI, можна побудувати стандартними методами [IETF RFC 2396]. Це здійснюється через скорочення в URI документі для документа, фрагмента чи сепаратора "#", і значення XML атрибуту типу ID.

## 7.2.5 GML колекції об'єктів

### 7.2.5.1 AbstractMemberType та похідні типи властивостей

Для створення GML колекції об'єктів, що не всі є географічними об'єктами, тип властивості має бути отриманий розширенням від `gml:AbstractMemberType`.

```
<complexType name="AbstractMemberType" abstract="true">
  <sequence/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Похідний тип властивості має відповідати одному з шаблонів, визначеним у пункті 7.2.3 та може встановлювати чисельність об'єктів у колекції, що потрібно для її використання за призначенням.



Абстрактний тип властивості призначений для використання лише в типах об'єктів, де програмне забезпечення має бути здатним виявити, що екземпляр такого об'єкту має бути інтерпретований як колекція об'єктів.

*Приклад 1.* Дивіться `gml:DictionaryEntryType` в пункті 15.2.3 для такого типу властивості.

За умовчанням такий абстрактний тип властивості не означає власність об'єктів у колекції. Атрибут `owns` із `gml:OwnershipAttributeGroup` можна використовувати на екземплярі елемента властивості для підтвердження власності об'єкту в колекції. Колекція не може володіти об'єктом, що вже перебуває у власності іншого об'єкту.

### 7.2.5.2 GML колекції об'єктів, `AggregationAttributeGroup`

GML колекція об'єктів є будь-яким `gml:AbstractObject` із елементом властивості у своїй моделі контенту, модель контенту якого отримана через розширення з `gml:AbstractMemberType`.

*Приклад 1.* `gml:Dictionary` є колекцією GML об'єктів, тому що модель контенту властивості `gml:dictionaryEntry`, визначена у пункті 15.2.3, отримана через розширення з `gml:AbstractMemberType`.

У додаток, комплексний тип, що описує модель контенту GML колекції об'єктів, може також містити посилання на групу атрибутів `gml:AggregationAttributeGroup` для подання додаткової інформації про семантику колекції об'єктів. Ця інформація може бути використана прикладними програмами для групування GML об'єктів, і (необов'язково) для їх упорядкування та індексації.

```
<attributeGroup name="AggregationAttributeGroup">
  <attribute name="aggregationType" type="gml:AggregationType"/>
</attributeGroup>
```

Допустимі значення для атрибуту `aggregationType` визначені через `gml:AggregationType`. Дивіться підрозділ 8.4 міжнародного стандарту ISO/IEC 11404:1996, де поданий смисл значень у переліку.

```
<simpleType name="AggregationType" final="#all">
  <restriction base="string">
    <enumeration value="set"/>
    <enumeration value="bag"/>
    <enumeration value="sequence"/>
    <enumeration value="array"/>
    <enumeration value="record"/>
    <enumeration value="table"/>
  </restriction>
</simpleType>
```

**Примітка 1.** Якщо колекцію типу агрегації “масив” реалізовано в прикладній схемі, тип масиву у прикладній схемі потребує додаткової інформації, щоб опрацювати індексацію.

**Примітка 2.** Якщо колекцію агрегації типу “таблиця” реалізовано в прикладній схемі, то табличний тип у прикладній схемі потрібен для моделювання додаткової інформації, щоб додати необхідну інформацію про поля та їх структуру.

## 7.2.6 Метадані

### 7.2.6.1 AbstractMetadataPropertyType

Для зв'язування метаданих, описаних XML схемою з GML об'єктом, треба визначити елемент властивості, модель контенту якого отримують через розширення від `gml:AbstractMetadataPropertyType`.

Значенням такої властивості будуть метадані. Модель контенту такого типу властивості, тобто прикладна схема метаданих, має бути визначена GML прикладною схемою:

```
<complexType name="AbstractMetadataPropertyType" abstract="true">
  <sequence/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Тип властивості, отриманий від `gml:AbstractMetadataPropertyType`, має відповідати одному з шаблонів, визначених для типів GML властивості в пункті 7.2.3.

За умовчанням абстрактні типи властивості не передбачають ніякої власності на дані. Атрибут `owns` групи `gml:OwnershipAttributeGroup` можна використовувати на екземплярі елементу властивості метаданих, щоб підтвердити власність на метадані. Якщо метадані, що відповідають концептуальній моделі міжнародного стандарту ISO 19115, мають бути закодовані в GML документі, то для кодування інформації метаданих треба користуватись відповідною специфікацією реалізації, визначеної в ISO/TS19139.

*Приклад 1.* Припустимо, що тип географічного об'єкту "Дорога" можна асоціювати з інформацією метаданих двох розділів – властивістю якості даних "horizontalAbsolutTochnist" та загальною властивістю "metadata" ("метадані") з ISO/TS 19139.

Це можна ввести в прикладну схему через об'єднання властивостей метаданих в одну комплексну властивість таким чином:

```
<complexType name="DorogaType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <!-- ... -->
        <element name="DorogaMetadata" type="ex:dDorogaMetadataPropertyType"/>
        <!-- ... -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<complexType name="DorogaMetadataPropertyType">
  <complexContent>
    <extension base="gml:AbstractMetadataPropertyType">
      <sequence>
        <element ref="ex:DorogaMetadata"/>
      </sequence>
    </extension>
  </complexContent>
</xs:complexType>
```

```

<complexType name="DorogaMetadataType">
  <complexContent>
    <extension base="gmx:AbstractObjectMetadata_Type">
      <sequence>
        <element name="horizontalAbsolutTochnist" minOccurs="0"
          type="gmd:DQ_AbsoluteZovnPozitsiyTochnist_PropertyType"/>
        <element name="metadata" minOccurs="0"
          type="gmd:MD_Metadata_PropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="DorogaMetadata" type="myAs:DorogaMetadataType"
  substitutionGroup="gmx:AbstractObjectMetadata"/>

```

Далі, екземпляр географічного *об'єкту* "Дорога Т" виглядатиме так:

```

<ex:Doroga>
  <!-- ... -->
  <ex:DorogaMetadata>
    <ex:dorogaMetadata>
      <ex:horizontalAbsolutTochnist>
        <gmd:DQ_AbsoluteZovnPotionalTochnist>
          <!-- The DQ_Element subelements are not detailed -->
        </gmd:DQ_AbsoluteZovnPotionalTochnist>
      </ex:horizontalAbsolutTochnist>
      <ex:metadata>
        <gmd:MD_Metadata>
          <!-- a full set of ISO/TS 19139 metadata elements -->
        </gmd:MD_Metadata>
      </ex:metadata>
    </ex:DorogaMetadata>
  </ex:dorogaMetadata>
  <!-- ... -->
</myAs:Doroga>

```

Альтернативним шляхом кодування, що подає властивості метаданих як окремих властивостей географічного об'єкту, є:

```

<complexType name="DorogaType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <!-- ... -->
        <element name="horizontalAbsolutTochnist" minOccurs="0">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractMetadataPropertyType">
                <sequence>
                  <element ref="gmd:DQ_AbsoluteZovnPotionalTochnist"/>
                </sequence>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ... -->
<element name="metadata" minOccurs="0">
  <complexType>
    <complexContent>
      <extension base="gml:AbstractMetadataPropertyType">
        <sequence>
          <element ref="gmd:MD_Metadata"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ... -->
</sequence>
</extension>
</complexContent>
</complexType>

```

Приклад екземпляру виглядав би таким чином:

```

<ex:Doroga>
  <!-- ... -->
  <ex:horizontalAbsolutTochnist>
    <gmd:DQ_AbsoluteZovnPostionalTochnist>
      <!-- The DQ_Element subelements are not detailed here -->
    </gmd:DQ_AbsoluteZovnPostionalTochnist>
  </ex:horizontalAbsolutTochnist>
  <!-- ... -->
  <ex:metadata>
    <gmd:MD_Metadata>
      <!-- a full set of ISO/TS 19139 metadata elements -->
    </gmd:MD_Metadata>
  </ex:metadata>
  <!-- ... -->
</ex:Doroga>

```

*Приклад 2.* Припустимо, набір даних має бути здатним містити метадані елементів про центральну частину Дубліна (Dublin Core). Їх можна внести в прикладну схему таким чином:

```

<xsd:import namespace="http://www.purl.org/dc/terms/schemaLocation="http://schemas.opengis.net/cs
w/2.0.0/recdcterms.xsd"/>

```

```

<complexType name="DatasetType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <!-- ... -->

```

```

        <element name="generalMetadata" type="ex:DublinCoreMetadataPropertyType"/>
        <!-- ... -->
    </sequence>
</extension>
</complexContent>
</complexType>

<complexType name="GeneralMetadataPropertyType">
    <complexContent>
        <extension base="gml:AbstractMetadataPropertyType">
            <sequence>
                <element ref="ex:DublinCoreMetadata"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="DublinCoreMetadata">
    <complexType name="DublinCoreMetadataType">
        <sequence>
            <group ref="dct:DCMI-terms" xmlns:dct="http://www.purl.org/dc/terms/" />
        </sequence>
    </complexType>
</element>

```

Приклад екземпляру виглядатиме так:

```

<ex:Dataset>
  <!-- ... -->
  <ex:generalMetadata>
    <ex:DublinCoreMetadata>
      <dc:title>Vector Smart Map Level 0</dc:title>
      <dc:abstract>Vector Map: a general purpose database design to support GIS
applications</dc:abstract>
      <dc:publisher>US National Geospatial-Intelligence Agency</dc:publisher>
      <dc:format>VPF</dc:format>
      <dc:coverage>world</dc:coverage>
      <dc:language>en</dc:language>
      <!-- ... -->
    </ex:DublinCoreMetadata>
  </ex:generalMetadata>
  <!-- ... -->
</ex:Dataset>

```

## 8 GML СХЕМА –XLINK ПОСИЛАННЯ ТА БАЗОВІ ТИПИ

### 8.1 Посилання – асоціації об'єктів та віддалені властивості

Нормативну специфікацію посилання Xlink надано від W3C [xlink].

**Примітка 1.** Документ схеми xlink.xsd подають як частину GML схеми в додатку В.

Компоненти Xlink використовують в GML для реалізації асоціацій між об'єктами через посилання. GML елементи властивості (див. пункт 7.2.3) можуть мати атрибути Xlink, які підтримують кодування асоціативних відношень через посилання, ім'я елемента властивості, що зазначає цільову роль в асоціації. Найважливішим компонентом посилання Xlink є:

xlink:href – ідентифікатор ресурсу, що є ціллю в асоціації, поданий як URI.

Поява xlink:href на GML властивості зазначає, що значення властивості має бути знайдене шляхом проходження посилання, тобто, значення вказане значенням атрибуту xlink:href. Відповідно до термінології Xlink, GML властивості з атрибутами xlink:href іноді називають віддаленими властивостями.

Інші компоненти Xlink вживають для зазначення додаткової семантики зв'язку. Найбільш корисними з них є:

- xlink:role – опис природи цільового ресурсу, поданого як URI;
- xlink:arcrole – опис ролі в асоціації або призначення цільового ресурсу у відношенні до поточного ресурсу, поданого як URI;
- xlink:title – опис асоціації чи цільового ресурсу, поданий як текст.

Повні визначення цих та інших Xlink компонентів, включно з їх використанням у розширених Xlink картах асоціацій, знаходяться в специфікації Xlink.

URI посилання визначають як необов'язковий вибір абсолютного чи відносного URI, після якого іде ідентифікатор фрагменту, що складається з решітки ("#") та інформації про додаткові посилання. Для властивостей та віддалених асоціацій GML об'єкту ця додаткова інформація про посилання має бути подана одним із наступних способів:

- shorthand (раніше назване "чисте ім'я") XPointer [XPointer Framework], що складається із значення атрибуту GML об'єкту або
- елементом () схеми на базі покажчика XPointer [XPointer element()], або
- схемою xpointer(), базованою на XPointer [XPointer xpointer()], що складається з виразу XPath [XPath], що вибирає GML object, та якому необов'язково передуює одна або більше xmlns() схем, базованих на покажчиках XPointer [XPointerxmlns()], що визначають префікси простору імен, використані у виразі XPath.

URI, що не містить абсолютного чи відносного URI, але повністю складається з ідентифікатора фрагменту, посилається на GML об'єкт десь ще в тому ж самому GML документі.

Абсолютні або відносні URI можуть містити компонент запиту, що складається із знаку запитання ("?"), що передуює запиту, щоб бути інтерпретованим через ресурс. Для властивостей та віддалених асоціацій GML об'єкту будь-який подібний запит має бути запитом до сервісу, що повертає GML об'єкт. В URI, що містить такий запит, можна (але необов'язково) вживати ідентифікатор фрагменту, залежно від синтаксису, визначеного сервісом.

У GML схемі прості посилання Xlink використовують виключно для вказування ролей GML об'єктів в асоціаціях та значень властивості, на які спрямовані віддалені посилання.

*Приклад 1.* Посилання на елемент об'єкту в тому самому документі GML може бути закодоване як :

```
<myProperty xlink:href="#o1"/>
```

*Приклад 2.* Посилання на елемент об'єкту у віддаленому документі XML, в якому використане значення gml:id цього об'єкту, закодоване як:

```
<myProperty xlink:href="http://my.big.org/test.xml#o1"/>
```

*Приклад 3.* Посилання на елемент об'єкту у віддаленому документі XML (або в сховищі GML об'єктів) із використанням значення властивості gml:identifier цього об'єкту можна закодувати як:

```
<myProperty xlink:href="http://my.big.org/test.xml#element
  (//gml:GeodeticCRS[./gml:identifier[@codeSpace="urn:ogc:def:crs:EPSG:6.3:"]="4326")
"/>
```

*Приклад 4.* Посилання на елемент об'єкту з уніформним іменем ресурсу (треба відзначити, що URN resolver потрібен для вирішення URN та доступу до об'єкту, на який спрямоване посилання):

```
<myProperty xlink:href="urn:x-ogc:def:crs:EPSG:6.3:4326"/>
```

Тип даних IDREF та унікальні, ключові елементи та елементи ключових посилань, визначені в специфікаціях XML та XML схем, забезпечують альтернативні механізми ідентифікації та зв'язування з типом даних ID та Xlink посилання для використання в окремому XML документі. Хоча ці XML компоненти можна вживати в XML схемах, вони не мають нормативної ролі в GML та не мають бути використані для указування ролі в асоціаціях GML об'єктів та значень властивості, на які спрямовані віддалені посилання.

## 8.2 Базові типи

### 8.2.1 Огляд

Схема W3C XML забезпечує набір “простих” вбудованих типів, що визначають методи подання значень як літерного тексту без внутрішньої розмітки (вони описані в W3C XML схемі, частина 2:2001). Оскільки GML є XML кодуванням, у якому екземпляри описані з використанням XML схеми, ці прості типи треба максимально використовувати для подання типів даних. W3C XML схема також забезпечує методи визначення:

- простих типів через обмеження та комбінації вбудованих типів та
- комплексних типів з простим контентом, але які мають атрибути XML.

У багатьох місцях, де зручних вбудованих простих типів немає в наявності, прості типи контенту, отримані з використанням механізмів XML схеми, вживають для подання типів даних у GML.

Набір цих простих типів контенту, що потрібні для кількох компонентів GML, визначено у схемі basicTypes, як і деякі елементи, базовані на них. Вони базовані головне навколо компонентів, потрібних для запису обсягів, кількостей, позначок та умов, разом із підтримкою виключень або нульових значень.

**Примітка 1.** Базові типи та елементи, описані в документі схеми basicTypes у додатку В. Схему ідентифіковано незалежним від розташування іменем (з використанням URN синтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:basicTypes:3.2.1
```

## 8.2.2 Зв'язок з ISO/TS 19103

В ISO/TS 19103 визначені базові типи для концептуальних схем в комплексі міжнародних стандартів ISO 19100. У GML реалізовано піднабір цих базових типів, описаний у додатку Г.2.2.

**Примітка 1.** Деякі з базових типів з ISO/TS 19103 визначені в інших документах GML схеми: одиниці міри визначені в розділі 16 та вектор у пункті 10.1.4.5.

## 8.2.3 Прості типи

### 8.2.3.1 NilReasonType

`gml:NilReasonType` визначає модель контенту, що забезпечує запис пояснення відсутності значення або іншого виключення.

```
<simpleType name="NilReasonEnumeration">
  <union>
<simpleType>
  <restriction base="string">
    <enumeration value="inapplicable"/>
    <enumeration value="missing"/>
    <enumeration value="template"/>
    <enumeration value="unknown"/>
    <enumeration value="withheld"/>
  </restriction>
</simpleType>
<simpleType>
  <restriction base="string">
    <pattern value="other:\w{2,}"/>
  </restriction>
</simpleType>
  </union>
</simpleType>

<simpleType name="NilReasonType">
  <union memberTypes="gml:NilReasonEnumeration anyURI"/>
</simpleType>
```

`gml:NilReasonType` є об'єднанням таких перелічуваних значень:

- "inapplicable" ("не підлягає застосуванню", значення відсутнє);
- "missing" ("пропущено", правильне значення не є готовим для постачальника даних, більше того, правильне значення може не існувати);
- "template" ("шаблон", значення буде наявним пізніше);
- "unknown" ("невідомо", правильне значення невідоме і його постачальник не обраховує його; однак, не виключено, що правильне значення існує);
- "withheld" ("затримано", значення не розголошується).
- "other:" ("інше": інші короткі пояснення, де текст є рядком з двох чи більше значень без включення пробілів);
- anyURI, що має зв'язок з ресурсом, що описує причину виключення.

У конкретних колах можуть обрати призначення більших обсягів детальної семантики для стандартних значень.



Певна спільнота може обрати призначення більш детальної семантики наданим стандартним значенням. Альтернативно, метод URI сприяє більш конкретному або повному поясненню відсутності значення, що має бути подане десь іще, та на яке мають бути спрямовані посилання у документі екземпляру.

`gml:nilReasonType` уживають як член об'єднання у числі простих типів контенту, що визначені далі (див. пункти 8.2.3.4, 8.2.4.1, 8.2.4.2, 8.2.4.3), де це потрібне для того, щоб дозволити використання значення з об'єднання `NilReasonType` як альтернативне для первинного типу.

### 8.2.3.2 Елементи, зазначені як такі, які можна обнулити

Атрибут схеми XML “такий, що можна обнулити” [W3C XSD] можна включити в декларацію будь-якого елемента в схемі.

**Примітка 1.** За умовчанням атрибут схеми “такий, що можна обнулити” (“nillable”) має значення “false” (“хибність”).

*Приклад 1.* В наступних декларації елементів показано використання атрибуту “nillable”:

```
<element name="amount" type="double" nillable="true"/>
<element ref="my:amount" nillable="true"/>
```

За декларацією елемента “таким, що можна обнулити”, екземпляр елемента може пропустити свій контент у випадках, де порожнє значення звичайно не діяло б у схемі через надання атрибуту `nil` із простору імен екземплярів у XML схемі із значенням “true” (“істина”).

*Приклад 2.* Елементи, що були задекларовані із цим `nillable="true"` у схемі можуть виступати в документах екземпляру як наступне:

```
<my:amount>34.567</my:amount>
<my:amount xsi:nil="true"/>
```

Декларування елемента як нуля є реалізацією типу даних “Void” (“Відсутнє”) з of ISO/IEC 11404, тобто є поданням “об’єкту, присутність якого є синтаксично або схематично потрібною, але не надає інформації в даному екземплярі” [ISO/IEC 11404].

**Примітка 1.** Це відрізняється від декларації елемента з атрибутом потужності, що встановлений для того, щоб зробити елемент необов’язковим, як наприклад:

```
<element name="amount" type="double" minOccurs="0"/>
```

що дозволяє цілком пропустити елемент в екземплярі.

В деяких ситуаціях, де треба зазначити елемент в прикладній схемі “як такий, що можна обнулити”, може бути зручним додання атрибуту типу `gml:nilReasonType`.

*Приклад 3.* Компоненти прикладної схеми

```
<element name="amount" nillable="true">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="nilReason" type="gml:nilReasonType"/>
      </extension>
    </simpleContent>
  </complexType>
</element>

<element name="money" type="my:NRMeasureType" nillable="true"/>
```

```
<complexType name="NRMeasureType">
  <simpleContent>
    <extension base="decimal">
      <attribute name="uom" type="token" use="required"/>
      <attribute name="nilReason" type="gml:NilReasonType"/>
    </extension>
  </simpleContent>
</complexType>
```

дозволили б доповнювати екземпляри з додатковим атрибутом, що пояснює відсутність значення, наприклад

```
<my:amount xsi:nil="true" nilReason="unknown"/>
<my:money xsi:nil="true" nilReason="other:myDaughterSpentIt" uom="AUD"/>
```

У GML схемі та у GML прикладних схемах, конструкції “nillable” та “nilReason” можна вживати для елементів, якими подано GML властивості (див. пункт 7.2.3). Це дозволяє властивостям бути частиною контенту об’єктів та географічних об’єктів у GML та GML прикладних мовах для того, щоб бути задекларованими як обов’язкові, у той же час із можливістю зустрічатися в документі екземпляру без значення.

**Примітка 1.** І прості, і комплексні елементи контенту можна декларувати як такі, що можна обнулити, тобто ця конструкція дозволяє єдиний синтаксис для властивостей із порожніми значеннями.

### 8.2.3.3 SignType

gml:SignType є типом зручності із значеннями “+” (плюс) and “-” (мінус):

```
<simpleType name="SignType">
  <restriction base="string">
    <enumeration value="-"/>
    <enumeration value="+"/>
  </restriction>
</simpleType>
```

**Примітка 1.** Елементи або атрибути цього типу використовують у різних місцях для вказування напрямку топологічних об’єктів з “+” для спрямованих уперед чи “-” для спрямованих назад.

### 8.2.3.4 booleanOrNilReason, doubleOrNilReason, integerOrNilReason, NameOrNilReason, stringOrNilReason

Типи gml:booleanOrNilReason, gml:doubleOrNilReason, gml:integerOrNilReason, gml:NameOrNilReason, gml:stringOrNilReason забезпечують розширення для відповідних убудованих простих типів XML схеми, що дозволяє вибір або значення вбудованого простого типу, або причини для відсутності значення. Вони побудовані таким чином:

```
<simpleType name="booleanOrNilReason">
  <union memberTypes="gml:NilReasonEnumeration boolean anyURI"/>
</simpleType>

<simpleType name="doubleOrNilReason">
  <union memberTypes="gml:NilReasonEnumeration double anyURI"/>
</simpleType>
```

```
<simpleType name="integerOrNilReason">
  <union memberTypes="gml:NilReasonEnumeration integer anyURI"/>
</simpleType>
```

```
<simpleType name="NameOrNilReason">
  <union memberTypes="gml:NilReasonEnumeration Name anyURI"/>
</simpleType>
```

```
<simpleType name="stringOrNilReason">
  <union memberTypes="gml:NilReasonEnumeration string anyURI"/>
</simpleType>
```

### 8.2.3.5 CodeType, CodeWithAuthorityType

`gml:CodeType` є узагальненим типом, що треба використовувати для термінів, ключових слів або імені:

```
<complexType name="CodeType">
  <simpleContent>
    <extension base="string">
      <attribute name="codeSpace" type="anyURI"/>
    </extension>
  </simpleContent>
</complexType>
```

Він додає XML атрибут `codeSpace` до терміну, де значення атрибуту `codeSpace` (якщо присутнє) має вказувати словник, тезаурус, схему класифікації, організацію або зразок для терміну.

*Приклад 1.* Схема `gmlBase` містить декларацію елемента з допомогою цього типу (див. 8.2.3.5)

```
<element name="name" type="gml:CodeType"/>
```

тому відповідний елемент можна зустріти в документі екземпляру у формі:

```
<gml:name codeSpace = "http://www.ukusa.gov/placenames">St Paul</gml:name>
```

цьому прикладі “St Paul” вказано як смислове ім’я згідно до <http://www.ukusa.gov/placenames>. Треба зауважити, що завжди правила для значень, включно з обмеженнями на унікальність, встановлені організацією, що відповідає за простір кодів `codeSpace`.

Похідний тип `gml:CodeWithAuthorityType` вимагає, щоб атрибут `codeSpace` був поданий у екземплярі.

```
<complexType name="CodeWithAuthorityType">
  <simpleContent>
    <restriction base="gml:CodeType">
      <attribute name="codeSpace" type="anyURI" use="required"/>
    </restriction>
  </simpleContent>
</complexType>
```

### 8.2.3.6 MeasureType, UomIdentifier

`gml:MeasureType` підтримує запис обсягів, закодованих як значення XML схеми типу “double”, разом із одиницями вимірювання, зазначеними атрибутом `uom`, скороченням для “units of measure” (“одиниці вимірювання”). Значення атрибуту `uom` ідентифікує референцну систему для обсягу, звичайно інтервальну чи пропорційну шкалу.

`gml:MeasureType` визначений так:

```
<complexType name="MeasureType">
  <simpleContent>
    <extension base="double">
      <attribute name="uom" type="gml:UomIdentifier" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

*Приклад 1.* Прикладна схема може містити декларацію елемента з використанням цього типу:

```
<element name="height" type="gml:MeasureType"/>
```

Елементи, що відповідають цьому, можуть зустрічатися в документі екземпляру даних у такій формі :

```
<height uom="m">1.4224</height>
<height uom="http://www.equestrian.org/units/hands">14</height>
```

де значення атрибуту `uom` ідентифікує одиницю вимірювання або ресурс, в якому визначена одиниця вимірювання.

Простий тип `gml:UomIdentifier` визначає синтаксис та простір значень ідентифікатора одиниці вимірювання.

Це є тип об’єднання, визначений таким чином:

```
<simpleType name="UomIdentifier">
  <union memberTypes="gml:UomSymbol gml:UomURI"/>
</simpleType>
```

Перший член типу об’єднання, `gml:UomSymbol`, визначений таким чином:

```
<simpleType name="UomSymbol">
  <restriction base="string">
    <pattern value="^[^:\n\r\t]+"/>
  </restriction>
</simpleType>
```

Цей тип визначає текстовий рядок з довжиною не менше одиниці та обмежений у тому, що не може містити такі знаки: “:” (двокрапка), “ ” (пробіл), (новий рядок), (повернення назад), (перехід). Це дозволяє значення, що відповідають відомим скороченням, таким як “kg”, “m/s” тощо.

**Примітка 1.** Рекомендовано, щоб символ був ідентифікатором для одиниці вимірювання, як визначено в “Unified Code of Units of Measure” (UCUM, “Уніфікований код одиниць вимірювання”) (<http://aurora.regenstrief.org/UCUM>). Це забезпечує набір символів та граматику

для побудови ідентифікаторів для одиниць вимірювання, що є унікальними та можуть бути легко введені з клавіатури, з підтримкою обмеженого набору знаків, відомого як 7-бітовий ASCII. ISO 2955 раніше забезпечував специфікацію в межах цієї області, але був відкликаний у 2001 році. UCUM значною мірою відповідає стандарту ISO 2955 із змінами для усунення неоднозначностей та інших труднощів.

Другий член типу об'єднання, `gml:UomURI`, визначений таким чином:

```
<simpleType name="UomURI">
  <restriction base="anyURI">
    <pattern value="([a-zA-Z][a-zA-Z0-9\-\+\.\.]*:\.\./\./#).*/>
  </restriction>
</simpleType>
```

Цей тип визначає URI, обмежений так, що він має починатися з однієї з таких послідовностей: “#”, “./”, “../”, або рядком знаків, після якого додається “:”. Ці шаблони надають упевненість, що підтримуються найбільш загальні URI форми, включно із абсолютними та відносними URI та URI, що є простими ідентифікаторами фрагментів, але забороняють певні форми, або відносний URI, які можна сплутати із символами одиниці вимірювання<sup>1)</sup>.

**Примітка 1.** Можна переписати такий відносний URI, щоб задовольнити обмеження (тобто, “./м/с”).

В документі екземпляр на елементах типу `gml:MeasureType` обов'язковий атрибут `uom` має містити значення, що відповідає:

- або умовному знаку одиниці вимірювання
- або посилання на визначення одиниці вимірювання, що не має умовного знаку, або коли є бажаним зазначити точне визначення або його варіант.

Компоненти GML для останньої мети визначені в підрозділі 16.2.

### 8.2.3.7 CoordinatesType

```
<complexType name="CoordinatesType">
  <simpleContent>
    <extension base="string">
      <attribute name="decimal" type="string" default="."/>
      <attribute name="cs" type="string" default=""/>
      <attribute name="ts" type="string" default="&#x20;/>
    </extension>
  </simpleContent>
</complexType>
```

Цей тип був скасований для кортежів з порядковими значеннями, що є числами.

`CoordinatesType` є текстовим рядком та призначений для використання при записі масиву кортежів або координат.

*Якщо неможливо провести внутрішню структуру рядку через перевірку схеми, деякі необов'язкові атрибути можуть бути надані в попередніх версіях GML задля підтримки опису внутрішньої структури. Ці атрибути скасовані. Атрибути були призначені для використання в такому вигляді:*

*Decimal* – символ, що використовують для десяткової крапки

<sup>1)</sup> наприклад, «м/с»

(default="." зупинка чи період)

*cs* – це символ, який використовують для розділення компонентів в складі кортежу або масиву координат:

(default="," кома)

*ts* – це символ, що використовують, щоб розділити кортежі або масиви кортежів:

(default=" " а пробіл)

Оскільки це базується на рядковому типі XML схеми, `gml:CoordinatesType` можна вживати в побудові таблиць кортежів або масивів кортежів, включно з тими, що містять сполучення текстових та числових значень.

Приклад 1. `<my:tupleList>bettong,357.,2.3 skink,140.,0.75 wombat,770.,17.5</my:tupleList>`

## 8.2.4 Списки

**8.2.4.1 booleanList, doubleList, integerList, NameList, NCNameList, QNameList, booleanOrNilReasonList, NameOrNilReasonList, doubleOrNilReasonList, integerOrNilReasonList**

Набір типів для списків простих значень побудований згідно до таких шаблонів:

```
<simpleType name="booleanList">
  <list itemType="boolean"/>
</simpleType>
```

```
<simpleType name="doubleList">
  <list itemType="double"/>
</simpleType>
```

```
<simpleType name="integerList">
  <list itemType="integer"/>
</simpleType>
```

```
<simpleType name="NameList">
  <list itemType="Name"/>
</simpleType>
```

```
<simpleType name="NCNameList">
  <list itemType="NCName"/>
</simpleType>
```

```
<simpleType name="QNameList">
  <list itemType="QName"/>
</simpleType>
```

```
<simpleType name="booleanOrNilReasonList">
  <list itemType="gml:booleanOrNilReason"/>
</simpleType>
```

```
<simpleType name="NameOrNilReasonList">
  <list itemType="gml:NameOrNilReason"/>
</simpleType>
```

```
<simpleType name="doubleOrNilReasonList">
  <list itemType="gml:doubleOrNilReason"/>
</simpleType>
```

```
<simpleType name="integerOrNilReasonList">
  <list itemType="gml:integerOrNilReason"/>
</simpleType>
```

Ці типи визначені як список значень убудованих простих типів відповідної XML схеми, або об'єднань типів, визначених у попередніх підкласах. Типи ...OrNilReasonList підтримують підстави для відсутності значень, розсіяних у межах списку.

**Примітка 1.** Ці типи подані як типи зручності. Вони можуть допомагати, коли треба визначити тип простого контенту, що є об'єднанням такого списку та другого типу простого контенту.

**Примітка 2.** Деякі типи починаються з великої літери, деякі – з малої літери. Підставою є те, що розмір початкової літери базового XML типу схеми збережений у GML типах для ясності.

**Примітка 3.** Елемент, у якому використовують один із цих типів, буде містити розділений пробілами список членів певного типу (див. <http://www.w3.org/TR/xmlschema-2/#atomic-vs-list> із більш детальним викладенням структури XML списку).

**Примітка 4.** У жодному із цих типів, визначених тут, не використовують рядок XML схеми як член списку. Підставою для цього є те, що рядок може містити вбудовані пробіли, розриви рядків тощо (<http://www.w3.org/TR/xmlschema-2/#string>). Оскільки пробіли діють як розділювальні знаки в екземплярі списку, виникає неоднозначність в ідентифікації членів, яка потенційно містить пробіл. З іншого боку, екземпляр типу у схемі XML імені може не містити пробілу (<http://www.w3.org/TR/2000/WD-xml-2e-20000814#NT-Name>), тому це можна безпечно використовувати в контексті списку. Висновок із цього є такий: якщо термін може містити пробіл, то цей термін не можна зустріти в екземплярі списку.

#### 8.2.4.2 CodeListType, CodeOrNilReasonListType

Два типи gml:CodeListType та gml:CodeOrNilReasonListType надають список термінів. Визначення схем є такими:

```
<complexType name="CodeListType">
  <simpleContent>
    <extension base="gml:NameList">
      <attribute name="codeSpace" type="anyURI"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<complexType name="CodeOrNilReasonListType">
  <simpleContent>
    <extension base="gml:NameOrNilReasonList">
      <attribute name="codeSpace" type="anyURI"/>
    </extension>
  </simpleContent>
</complexType>
```

Значення в елементі екземпляру типу списку `gml:CodeListType` має бути дійсним згідно до правил словника, схеми класифікації, організації, ідентифікованої за значенням її атрибуту `codeSpace`.

*Приклад 1.* Прикладна схема може містити декларацію елемента з використанням цього типу

```
<element name = "species" type = "gml:CodeListType"/>,
тому відповідний елемент може зустрічатися в документі екземпляру таким чином:
<species codeSpace="http://my.big.org/florelegium">dryandra banksia hardenbergia
lavendar</species>
```

де перелічені члени списку, взяті з <http://my.big.org/florelegium>, що є (гіпотетично) списком видів квітів.

Елемент екземпляру типу `gml:CodeOrNilReasonListType` може також містити вбудовані значення з типу `NilReasonType`. Це призначено для використання в ситуаціях, де очікується термін або класифікація, але значення може бути відсутнє з деяких причин.

### 8.2.4.3 MeasureListType, MeasureOrNilReasonListType

Два типи `gml:MeasureListType` та `gml:MeasureOrNilReasonListType` надані для списків величин. Схема визначена таким чином:

```
<complexType name="MeasureListType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attribute name="uom" type="gml:UomIdentifier" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="MeasureOrNilReasonListType">
  <simpleContent>
    <extension base="gml:doubleOrNilReasonList">
      <attribute name="uom" type="gml:UomIdentifier" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

*Приклад 1.* Прикладна схема може містити декларації елементів із використанням цих типів.

```
<element name = "heights" type = "gml:MeasureListType"/>
<element name = "weights" type = "gml:MeasureOrNilReasonListType"/>
```

тому відповідні елементи можуть зустрічатися в документі екземпляру таким чином:

```
<heights uom="m">1.76 1.85 1.56 1.98</heights>
<weights uom="kg">67.0 73.4 withheld 85.1</weights>
```

В обох прикладах усі значення в списку описані з використанням такої ж шкали.

У другому прикладі значення, що описує підставу для порожнього значення, міститься там, де звичайно очікується міра, але значення може бути відсутнім з деякої причини.



## 9 GML СХЕМА – ГЕОГРАФІЧНІ ОБ'ЄКТИ

### 9.1 Загальні концепції

GML географічний об'єкт є географічним об'єктом, закодованим з використанням GML.

*Приклад 1.* Дорога, річка, особа, засіб пересування, адміністративна одиниця, подія і т. і.

Схема географічного об'єкту надає структуру для створення GML географічних об'єктів та колекцій географічних об'єктів.

**Примітка 1.** Документ схеми географічних об'єктів feature.xsd (див. додаток В) визначений через таке, незалежне від розміщення, ім'я (з допомогою синтаксису URN):

urn:ogc:specification:gml:schema-xsd:feature:3.2.1

### 9.2 Зв'язок з ISO 19109

Модель GML географічних об'єктів відповідає принципам, викладеним в ISO 19109:2005, розділ 7. Це забезпечує узгоджену часткову реалізацію загальної моделі географічних об'єктів у ISO 19109. Цей зв'язок докладно описане в додатку Г.2.6.

**Примітка 1.** Модель географічних об'єктів також відображає концепцію колекції географічних об'єктів з абстрактної специфікації OGC, тема 5 і 10.

### 9.3 Об'єкти

#### 9.3.1 AbstractFeatureType

Базову модель географічних об'єктів подають через gml:AbstractFeatureType, визначений у схемі так:

```
<complexType name="AbstractFeatureType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:boundedBy" minOccurs="0"/>
        <element ref="gml:location" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Модель контенту для gml:AbstractFeatureType додає дві специфічні властивості, що підходять для географічних об'єктів, до моделі контенту, визначеній в gml:AbstractGMLType.

Значення властивості gml:boundedBy описує обмежувальний прямокутник, що містить повний екземпляр об'єктів та є корисним головне для підтримки швидкого пошуку об'єктів, що приходяться на певну позицію.

Значення властивості gml:location описує екстент, позицію та відносне розташування об'єкту.

*gml:location* було скасовано як частину стандартної моделі контенту gml:AbstractFeatureType.

#### 9.3.2 AbstractFeature

Елемент gml:AbstractFeature задекларований таким чином:

```
<element name="AbstractFeature" type="gml:AbstractFeatureType" abstract="true"
  substitutionGroup="gml:AbstractGML"/>
```

Абстрактний елемент виступає як головний у групі заміни, що може містити будь-які елементи, модель контенту яких отримують від `gml:AbstractFeatureType`. Його можна вживати як змінну в побудові моделей контенту.

`gml:AbstractFeature` можна вважати як “будь-що, що є GML географічним об’єктом”, та можна використовувати для визначення змінних або шаблонів, у яких значенням властивості GML є “будь-який географічний об’єкт”. Він зустрічається зокрема в колекції об’єктів GML (див підрозділ 9.9), де властивості географічного об’єкту, що до неї входить, містять одну чи кілька копій `gml:AbstractFeature` відповідно.

## 9.4 Стандартні властивості об’єкту

### 9.4.1 `boundedBy`, `BoundingShapeType`, `EnvelopeWithTimePeriod`, `EnvelopeWithTimePeriodType`

Ця властивість описує мінімальний обмежувальний прямокутник, що обмежує цілий географічний об’єкт. Його модель контенту є такою:

```
<element name="boundedBy" type="gml:BoundingShapeType" nillable="true"/>

<complexType name="BoundingShapeType">
  <sequence>
    <choice>
      <element ref="gml:Envelope"/>
      <element ref="gml:Null"/>
    </choice>
  </sequence>
  <attribute name="nilReason" type="gml:NilReasonType"/>
</complexType>
```

Елемент `gml:Envelope` визначений в пункті 10.1.4.6.

Відсутнє значення можна закодувати, як описано в пункті 8.2.3.2. Атрибут `nilReason` можна використовувати в таких випадках для відсутнього значення.

*Значення `gml:Null`, використане у попередніх версіях GML для кодування відсутнього значення, що не може бути застосоване або відсутнє за певної причина, скасовано.*

**Примітка 1.** Оскільки обмежувальний прямокутник визначений просто через позиції двох діагонально протилежних кутів, точний слід обмежувального прямокутника залежить від референцної системи координат, що застосована. Якщо описаний географічний об’єкт має нульовий екстент, то ці два кути співпадають, та обмежувальний прямокутник має нульові розміри. Властивість `boundedBy` надана постачальником даних для зручності. Значення обмежувального прямокутника звичайно обчислює споживач даних із просторово-часових властивостей географічного об’єкту. Як і для всіх властивостей, це лежить на відповідальності постачальника даних, щоб забезпечити правильність значення.

Для обмежувальних прямокутників, що включають часовий екстент, подають `gml:EnvelopeWithTimePeriod`, визначений як:

```
<element name="EnvelopeWithTimePeriod" type="gml:EnvelopeWithTimePeriodType"
substitutionGroup="gml:Envelope"/>

<complexType name="EnvelopeWithTimePeriodType">
  <complexContent>
    <extension base="gml:EnvelopeType">
```

```

<sequence>
  <element name="beginPosition" type="gml:TimePositionType"/>
  <element name="endPosition" type="gml:TimePositionType"/>
</sequence>
<attribute name="frame" type="anyURI" default="#ISO-8601"/>
</extension>
</complexContent>
</complexType>

```

Це додає дві властивості позиції в часі (початку і кінця), gml:beginPosition та gml:endPosition, що описують екстент часового обмежувального відрізка.

Оскільки gml:EnvelopeWithTimePeriod призначений для групи заміни, головою якої є gml:Envelope, його можна використовувати завжди, коли є дійсним gml:Envelope.

**Примітка 2.** Звичайно з усіма елементами геометрії, отриманими з gml:AbstractGeometryType (див. підпункт 10.1.3.1), референцні системи координат, що використані для позицій, які визначають gml:Envelope, можна вказати з допомогою необов'язкового XML атрибуту srsName. Якщо референцна система координат містить часову вісь, то gml:Envelope можна використати для опису просторово-часового екстенту.

#### 9.4.3 locationName, locationReference

Елемент властивості gml:locationName є властивістю для зручності, де текстове значення описує місцезположення. Він визначений так:

```
<element name="locationName" type="gml:CodeType"/>
```

Якщо імена місцезположення вибрані з контрольованого списку, то список треба ідентифікувати в атрибуті codeSpace.

Елемент властивості gml:locationReference є властивістю для зручності, де текстове значення, на яке спрямоване посилання з допомогою атрибуту xlink:href, описує місцезположення географічного об'єкту. Він визначений так:

```
<element name="locationReference" type="gml:ReferenceType"/>
```

*Приклад 1.* У наступних випадках показано різні шляхи, по яких locationName або locationReference можуть з'явитися в екземплярі даних.

Місцезположення, подане з допомогою імені з контрольованого джерела:

```

<Feature>
  <gml:locationName
codeSpace="http://www.icsm.gov.au/icsm/cgna/index.html">Leederville</gml:locationName>
</Feature>

```

Місцезположення, подане з допомогою текстового рядку:

```

<Feature>
  <gml:locationName>Nigel Foster's town of residence</gml:locationName>
</Feature>

```

Місцезположення, подане іншою службою:

```

<Feature>
  <gml:locationReference
xlink:href="http://www.ga.gov.au/bin/gazm01?placename=leederville&placetype=R&state=WA+"/>
</Feature>

```

### 9.4.3 FeaturePropertyType, FeatureArrayType

Особливий клас властивостей визначає асоціації між географічними об'єктами. У них використовують шаблон `gml:AssociationRoleType` таким чином:

```
<complexType name="FeaturePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractFeature"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Буває корисним визначати властивість, що містить масив інших географічних об'єктів. Це роблять з використанням типу властивості масиву географічних об'єктів, як визначено в наступній моделі контенту:

```
<complexType name="FeatureArrayType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractFeature" />
  </sequence>
</complexType>
```

### 9.5 Властивості геометрії

Імена, специфічні для аплікації, треба обирати для властивостей геометрії в GML прикладних схемах. Імена властивостей мають бути підібрані так, щоб відображати семантику значення. Використання специфічних імен є переважним методом для імен властивостей, включно із властивостями геометрії.

Немає успадкованих обмежень у типі властивості геометрії, що може мати географічний об'єкт, доки значенням властивості є об'єкт геометрії, яким можна замінити `gml:AbstractGeometry`.

*Приклад 1.* Тип географічного об'єкту “радіоретранслятор” міг би мати місцезнаходження, що повертає точкову геометрію для ідентифікації місцезнаходження через репрезентативну точку, та мати іншу властивість геометрії, названу `floorSpace`, що повертає поверхневу геометрію, яка описує його фізичну структуру, а також третю властивість геометрії, названу `serviceArea`, що повертає поверх нову геометрію, яка описує територію, в межах якої передачі можуть бути надійно отримані:

```
<complexType name="RadioTowerType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="location" type="gml:PointPropertyType"/>
        <element name="floorSpace" type="gml:SurfacePropertyType"/>
        <element name="serviceArea" type="gml:SurfacePropertyType"/>
        <!-- ... -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

GML схема включає наперед визначені типи властивості, які можна використовувати як типи елементу властивості геометрії.

**Таблиця 4** – Наперед визначені типи властивостей геометрії

Тип властивості в XML схемі	Асоційовані типи об'єктів геометрії (імена елементів)
PointPropertyType	Point
CurvePropertyType	AbstractCurve LineString Curve OrientableCurve CompositeCurve
SurfacePropertyType	AbstractSurface Polygon Surface OrientableSurface CompositeSurface
SolidPropertyType	AbstractSolid Solid CompositeSolid
MultiPointPropertyType	MultiPoint
MultiCurvePropertyType	MultiCurve
MultiSurfacePropertyType	MultiSurface
MultiSolidPropertyType	MultiSolid
MultiGeometryPropertyType	MultiGeometry
PointArrayPropertyType	Point(s)
CurveArrayPropertyType	AbstractCurve(s) LineString(s) Curve(s) OrientableCurve(s) CompositeCurve(s)
SurfaceArrayPropertyType	AbstractSurface(s) Polygon(s) Surface(s) OrientableSurface(s) CompositeSurface(s)
SolidArrayPropertyType	AbstractSolid(s) Solid(s) CompositeSolid(s)

### 9.6 Властивості топології

Як і у випадку з властивостями геометрії, імена, специфічні для аплікації, треба вибирати для властивостей топології в GML прикладних схемах. Імена властивостей треба вибирати для вираження семантики значення.

*Приклад 1.* Тип об'єктів *StatisticalArea* міг би мати одну чи кілька властивостей межі *boundary*, що повертають *TopoCurve* для подання межі статистичної території, та одну чи більше

властивостей *surface*, що повертають TopoSurface що повертають статистичну територію як таку.

```
<complexType name="StatisticalAreaType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="boundary" type="gml:TopoCurvePropertyType" maxOccurs="unbounded"/>
        <element name="surface" type="gml:TopoSurfacePropertyType" maxOccurs="unbounded"/>
        <!-- ... -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

GML схема включає наперед визначені типи властивості, які можна використовувати як елемент властивості топології. Перші чотири з цих властивостей виражають напрямок, на відміну від інших.

**Таблиця 5** – Наперед визначені формальні типи властивості топології

Схема типу властивості в XML	Асоційовані типи об'єктів топології (імена елементів)
DirectedNodePropertyType	Node
DirectedEdgePropertyType	Edge
DirectedFacePropertyType	Face
DirectedTopoSolidPropertyType	TopoSolid
TopoPointPropertyType	TopoPoint
TopoCurvePropertyType	TopoCurve
TopoSurfacePropertyType	TopoSurface
TopoVolumePropertyType	TopoVolume
TopoComplexMemberType	TopoComplex

### 9.7 Часові властивості

Подібно до властивостей геометрії та топології, за визначення часових властивостей відповідальний розробник прикладної схеми.

*Приклад 1.* Тип географічного об'єкту “будівля” може мати властивості constructionTime, XML тип якої є gml:TimePeriodPropertyType”, completionTime (час побудови), XML тип якої є “gml:TimeInstantPropertyType”, та age (вік), XML тип якої є “duration” (тривалість), або “gml:TimeIntervalLengthType”.

```
<complexType name="BuildingType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="constructionTime" type="gml:TimePeriodPropertyType"/>
        <element name="completionTime" type="gml:TimeInstantPropertyType"/>
        <element name="age" type="gml:TimeIntervalLengthType"/>
        <!-- ... -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

</complexContent>  
</complexType>

Ці типи показані в таблиці 6 та призначені для прямого використання в декларуванні елементів властивості. Слід зауважити, що жодні елементи не були наперед визначені для більшості типів, оскільки ці типи не використовують у GML схемі як такі.

**Таблиця 6** – Наперед визначені формальні типи часових властивостей

Тип властивості в XML схемі	Асоційовані типи об'єктів топології (імена елементів)
TimePrimitivePropertyType	AbstractTimePrimitive AbstractTimeGeometricPrimitive TimeInstant TimePeriod AbstractTimeTopologyPrimitive TimeEdge TimeNode
TimeGeometricPrimitivePropertyType	AbstractTimeGeometricPrimitive TimeInstant TimePeriod
TimeInstantPropertyType	TimeInstant
TimePeriodPropertyType	TimePeriod
TimeTopologyPrimitivePropertyType	AbstractTimeTopologyPrimitive TimeEdge TimeNode
TimeEdgePropertyType	TimeEdge
TimeNodePropertyType	TimeNode
TimeTopologyComplexPropertyType	TimeTopologyComplex
TimeOrdinalEraPropertyType	TimeOrdinalEra
TimeCalendarPropertyType	TimeCalendar
TimeCalendarEraPropertyType	TimeCalendarEra
TimeClockPropertyType	TimeClock
TimePositionType	- (simple type)
xsd:duration	- (simple type)
TimeIntervalLengthType	- (simple type)

Елементи часових властивостей та типи, перелічені вище, забезпечують відносно повний набір компонентів для асоціювання часової інформації з географічними та іншими об'єктами.

### 9.8 Визначення специфічно-прикладних типів географічних об'єктів

Всі специфічні типи географічних об'єктів, визначені в прикладних схемах, треба реалізовувати як глобальні XML елементи, модель контенту яких (типи XML схеми) отримують від `gml:AbstractFeatureType`, і тому всі GML географічні об'єкти успадковують необов'язкову властивість `gml:boundedBy` так же, як і стандартні властивості `gml:identifier`, `gml:description`, `gml:descriptionReference` та `gml:name`,

успадковані в свою чергу від `gml:AbstractGMLType`, якщо кожна властивість не відкинута у походженні через обмеження. `gml:AbstractFeatureType` також успадковує `gml:id` від `gml:AbstractGMLType`, і це є переважним засобом підтримки ідентифікаторів баз даних у GML.

**Примітка 1.** Скасовані властивості можна пропустити в списку успадкованих властивостей.

Це отримання типів вимагає засобів для того, щоб програмне забезпечення загального призначення, що може обробляти довільні дані, було здатним до проходження по дереву походження XML схеми для визначення, чи є даний елемент у потоці даних GML географічний об'єктом.

GML географічний об'єкт має набір властивостей, де специфічний набір властивостей визначає тип об'єкту. Властивості мають прості значення з допомогою простих типів контенту XML схеми, або властивості можуть мати комплексні значення, і тоді їх треба декларувати з допомогою шаблонів, описаних у пункті 7.2.3.

У прикладній схемі, що визначає географічний об'єкт, треба декларувати глобальний елемент, ім'я якого є семантичним типом географічного об'єкту в домені предметної області. Глобальний елемент має бути членом групи заміни `gml:AbstractFeature` (прямо чи непрямо).

```
<element name="<<featureName>>" type = "<<contentModel >>" substitutionGroup="gml:AbstractFeature" />
```

Модель контенту географічного об'єкту має бути іменована чи мати комплексний анонімний тип.

## 9.9 Колекції об'єктів

### 9.9.1 GML колекції географічних об'єктів

GML колекція географічних об'єктів є колекцією екземплярів об'єктів у GML.

GML колекція географічних об'єктів є GML географічним об'єктом із елементом властивості в його моделі контенту, модель контенту якого отримана через розширення з `gml:AbstractFeatureMemberType` (див. пункт 9.9.2).

У додаток, комплексний тип, що описує модель контенту колекції GML географічних об'єктів, може також включати посилання на групу атрибутів `gml:AggregationAttributeGroup` для подання додаткової інформації про семантику колекції об'єктів, як це визначено в підпункті 7.2.5.1.

*Приклад 1.* Наступні компоненти схеми моделюють просту колекцію об'єктів:

```
<element name="MyFeatures" type="ex:MyFeaturesType"
substitutionGroup="gml:AbstractFeature"/>
<complexType name="MyFeaturesType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="myMember" type="ex:MyFeaturesMemberType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```



```

<complexType name="MyFeaturesMemberType">
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="gml:AbstractFeature"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

Приклад екземпляру кодування колекції семантиками набору, де забезпечено обмежувальний прямокутник:

```

<MyFeatures aggregationType="set">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
      <gml:lowerCorner>50.23 9.23</gml:lowerCorner>
      <gml:upperCorner>50.31 9.27</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <myMember>
    <MyFeature gml:id="f1"/>
  </myMember>
  <myMember>
    <MyFeature gml:id="f2"/>
  </myMember>
  <myMember xlink:href="#f3"/>
</MyFeatures>

```

*Приклад 2.* Часто колекція об'єктів містить екземпляри специфічного типу. У наступному прикладі колекцією географічних об'єктів є дорога (road), що складається з відрізків.

```

<element name="Road" type="ex:RoadType" substitutionGroup="gml:AbstractFeature"/>
<complexType name="RoadType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="segment" type="ex:RoadMemberType" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="RoadMemberType">
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">

```

```

    <sequence minOccurs="0">
      <element ref="ex:RoadSegments"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </extension>
</complexContent>
</complexType>

```

Далі – фрагмент прикладу, що кодує упорядковану колекцію відрізків дороги:

```

<Road gml:id="r1" aggregationType="sequence">
  <segment>
    <RoadSegment gml:id="s1"/>
  </segment>
  <segment xlink:href="#s8"/>
  <segment>
    <RoadSegment gml:id="s4"/>
  </segment>
</Road>

```

### 9.9.2 AbstractFeatureMemberType та похідні типи властивості

Для створення колекції географічних об'єктів треба отримати тип властивості через розширення від AbstractFeatureMemberType.

```

<complexType name="AbstractFeatureMemberType" abstract="true">
  <sequence/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Похідна властивість має відповідати одному із шаблонів, визначених у пункті 7.2.3 та може встановлювати чисельність об'єктів, як це потрібно для її цільового використання.

За умовчанням, абстрактний тип властивості не означає ніякої власності на географічні об'єкти в колекції. Атрибут owns із групи gml:OwnershipAttributeGroup можна вживати як елемент екземпляру властивості для того, щоб підтвердити власність на географічний об'єкт у колекції. Колекції не може належати географічний об'єкт, що знаходиться у власності іншого об'єкту.

### 9.10 Просторова референсна система, використана в географічних об'єктах або колекціях географічних об'єктів

Значення властивості gml:boundedBy для географічного об'єкту або колекції географічних об'єктів звичайно є gml:Envelope. Як правило, з усіма елементами геометрії, отриманими від gml:AbstractGeometryType (див. підпункт 10.1.3.1), референсну систему координат, використану для визначення позицій, що визначають gml:Envelope, можна указати з допомогою XML атрибуту srsName.

Для зручності побудови географічного об'єкту або екземплярів колекцій географічних об'єктів, значення атрибуту srsName на gml:Envelope, що є значенням властивості gml:boundedBy географічного об'єкту, має бути успадковане через безпосередньо подані геометрії у всіх властивостях географічного об'єкту або членів колекції, якщо це не заперечено присутністю локального srsName. Тому необов'язково

для геометрії містити атрибут `srsName`, якщо в ній використана та ж сама референсна система координат, що і подана у властивості `gml:boundedBy` цього батьківського географічного об'єкту. Успадковування референцних систем координат продовжується до будь-якої глибини вкладання, але, якщо це пере визначено локальною декларацією `srsName`, то нова референсна система координат буде успадкована всіма її нащадками.

Всі геометрії, використані в географічному об'єкті або колекції географічних об'єктів, можуть мати атрибути `srsName` з тим, щоб зазначити референцну систему, використану локально, навіть якщо вона та ж, що і у батьківського об'єкту.

## 10 GML СХЕМА– ГЕОМЕТРИЧНІ ПРИМІТИВИ

### 10.1 Загальні концепції

#### 10.1.1 Огляд

**Примітка 1.** Модель геометрії у GML узгоджена із міжнародним стандартом ISO 19107. Базові концепції типів та елементів GML моделі геометрії розглянуто в цьому документі більш докладно.

У цьому розділі описані компоненти для геометричних примітивів, як це визначено в GML.

**Примітка 2.** Відповідні документи схем геометрії `geometryBasic0d1d.xsd` та `geometryBasic2d.xsd` (див. у додатку В), ідентифіковані такими незалежними від розташування іменами (з допомогою URN синтаксису):

`urn:x-ogc:specification:gml:schema-xsd:geometryBasic0d1d:3.2.1`

`urn:x-ogc:specification:gml:schema-xsd:geometryBasic2d:3.2.1`

`urn:x-ogc:specification:gml:schema-xsd:geometryPrimitives:3.2.1`

Будь-який елемент, що успадковує семантику від `AbstractGeometryType`, можна розглядати як множину прямих позицій.

Всі класи, отримані з `AbstractGeometryType`, успадковують необов'язкову асоціацію з референчною системою координат. Всі прямі позиції мають бути прямо чи непрямо асоційовані з референчною системою координат. Коли елементи геометрії агреговані в інші елементи геометрії (такі як `gml:MultiGeometry` або `gml:GeometricComplex`), що вже мають визначену референцну систему координат, ці елементи вважають як такі, що мають бути в тій же референційній системі координат, якщо не визначено інакше.

У моделі геометрії розрізняють геометричні примітиви, агрегати та комплекси.

Геометричні примітиви, тобто екземпляри підтипу `AbstractGeometricPrimitiveType`, будуть відкритими, тобто, вони не міститимуть точок своїх меж; криві не міститимуть своїх кінцевих точок, поверхні не міститимуть ліній меж, а тіла не міститимуть своїх обмежувальних поверхонь.

#### 10.1.2 Зв'язок з ISO 19107

Компоненти просторової геометрії GML схеми, визначені в розділах 10 та 11, забезпечують узгоджену часткову реалізацію просторової схеми (геометрії) ISO 19107. Цей зв'язок докладно розглянуто в додатку Г.2.3.

Впроваджені в GML типи геометрії визначені в ISO 19107; деякі додаткові обмеження визначені в ISO 19107 для цих типів, що є також обмеженнями на компоненти просторової геометрії GML схеми.

У додатку Г.3.5 також описані компліментарні компоненти схеми просторової геометрії, визначені в GML

### 10.1.3 Абстрактна геометрія

#### 10.1.3.1 AbstractGeometryType

```
<complexType name="AbstractGeometryType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
    </extension>
  </complexContent>
</complexType>
```

Всі елементи геометрії, отримані прямо чи непрямо з цього абстрактного надтипу. Елемент геометрії може мати ідентифікаційний атрибут (gml:id), одне чи більше імен (елементи gml:identifier та gml:name) та опис (елементи gml:description і gml:descriptionReference)<sup>2)</sup>. Це можна асоціювати з просторовою референцною системою (група атрибутів gml:SRSReferenceGroup).

Треба дотримуватись таких правил:

- кожний тип геометрії має походити від цього абстрактного типу;
- кожен елемент геометрії (тобто, елемент або тип геометрії) має бути прямо чи непрямо в групі заміни AbstractGeometry.

#### 10.1.3.2 SRSReferenceGroup

```
<attributeGroup name="SRSReferenceGroup">
  <attribute name="srsName" type="anyURI" />
  <attribute name="srsDimension" type="positiveInteger" />
  <attributeGroup ref="gml:SRSInformationGroup"/>
</attributeGroup>
```

Група атрибутів SRSReferenceGroup є обов'язковим посиланням на референцну систему координат, використану цією геометрією, з обов'язковою додатковою інформацією для спрощення обробки координат, коли більш докладне визначення референцної системи координат непотрібне.

В цілому, атрибут srsName вказує на екземпляр референцної системи координат з gml:AbstractCoordinateReferenceSystem (див. пункт 12.2.3). Для широко відомих посилань не є обов'язковим існування визначення референцної системи координат в місці, на яке вказує URI.

Якщо жоден атрибут srsName не надано, то референцну систему координат треба визначити як частину ширшого контексту, частиною якого є цей елемент геометрії.

*Приклад 1.* Геометрична агрегація або колекція об'єктів є типовими “ширшими контекстами”.

**Примітка 1.** Ім'я “srsName” було обране довільно. У поточній версії GML “crsName” має бути більш доречним, однак, у майбутніх версіях інші типи просторових референцних систем, тобто ті, в яких використовують географічні ідентифікатори, також можуть підтримуватися в GML.

<sup>2)</sup> Скасовані властивості пропущені в цьому списку. Тим не менш, вони досі залишаються дійсним контентом

Необов'язковий атрибут `srDimension` є числом значень координат позиції. Ця розмірність отриманий від референцної системи координат. Коли атрибут `srName` пропущено, цей атрибут також можна пропустити.

### 10.1.3.3 SRSInformationGroup

```
<attributeGroup name="SRSInformationGroup">
  <attribute name="axisLabels" type="gml:NCNameList" />
  <attribute name="uomLabels" type="gml:NCNameList" />
</attributeGroup>
```

Атрибути `uomLabels` та `axisLabels`, визначені в групі атрибутів `gml:SRSInformationGroup`, є необов'язковою додатковою надлишковою інформацією про РСК для спрощення процесу обробки значень координат, коли не потрібне більш повне визначення РСК. Ця інформація буде тією ж, що і включена в повне визначення РСК, на яку посилається атрибут `srName`. Коли атрибут `srName` включено, то або обидва атрибути `uomLabels` та `axisLabels` треба включати, або жоден з них не можна включати. Якщо пропущено атрибут `srName`, то обидва ці атрибути також треба пропустити.

Атрибут `axisLabels` є впорядкованим списком написів імен для всіх осей даної референцної системи координат. Значення `gml:axisAbbrev` треба використовувати для всіх цих написів, після усунення пробілів та недопустимих знаків. Коли включений атрибут `srName`, цей атрибут є необов'язковим. Коли пропущено атрибут `srName`, то цей атрибут також треба пропустити.

Атрибут `uomLabels` є впорядкованим списком написів одиниць вимірювання для всіх осей цієї референцної системи координат. Значення рядку в `gml:catalogSymbol` треба використовувати для цих написів одиниць вимірювання, після усунення пробілів та недопустимих знаків. Коли атрибут `axisLabels` включений, цей атрибут також треба включати. Якщо пропущено атрибут `axisLabels`, то цей атрибут також треба пропустити.

### 10.1.3.4 AbstractGeometry

```
<element name="AbstractGeometry" type="gml:AbstractGeometryType" abstract="true"
substitutionGroup="gml:AbstractGML" />
```

Елемент `AbstractGeometry` є абстрактним головою групи заміни для всіх GML елементів геометрії. Він включає наперед визначені та визначені користувачем елементи геометрії. Будь-який елемент геометрії має бути прямим чи непрямым розширенням чи обмеженням для типу `AbstractGeometryType` та має прямо чи непрямо бути в групі заміни `gml:AbstractGeometry`.

У додатку Г.2.3.2 визначена реалізація для геометричного об'єкту `GM_Object` (ISO 19107) з допомогою цього GML об'єкту.

### 10.1.3.5 GeometryPropertyType

```
<complexType name="GeometryPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeometry"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Геометрична властивість може бути або будь-яким елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалені включають елементи геометрії, розміщені десь іще в поточному чи іншому документі). Слід зауважити, що треба обов'язково подати елемент, що міститься, або посилання на нього, але не перше та друге разом (див. пункт 7.2.3).

Якщо об'єкт має властивість, що приймає геометрію як своє значення, то її називають властивістю геометрії. Загальним типом для такої властивості геометрії є `gml:GeometryPropertyType`, що відповідає загальними правилам, описаним у пункті 7.2.3.

#### 10.1.3.6 GeometryArrayPropertyType

```
<complexType name="GeometryArrayPropertyType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractGeometry" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Якщо географічний об'єкт має властивість, що приймає масив елементів геометрії як своє значення, це називається властивістю масиву геометрії. Загальним типом для такої властивості геометрії є `gml:GeometryArrayPropertyType`, який відповідає загальним правилам, описаним у пункті 7.2.3.

Елементи завжди включені в значенні масиву; посилання на елементи чи масиви геометрії через XLinks не підтримуються.

*Приклад 1.* Всі елементи в `gml:GeometryArrayPropertyType` є елементами типу `gml:AbstractGeometryType` (включно з типами, отриманими з цього абстрактного базового типу), доки елемент може прямо чи непрямо заміщати `gml:AbstractGeometry`.

### 10.1.4 Координатна геометрія, вектори та обмежувальні прямокутники

#### 10.1.4.1 DirectPositionType, pos

```
<complexType name="DirectPositionType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<sch:pattern name="CRS attributes constraints">
  <sch:rule id="CRSLabelsPos">
    <sch:report test="not(@srsDimension) or @srsName">Присутність атрибуту
розмірності означає присутність атрибуту srsName attribute.</sch:report>
    <sch:report test="not(@axisLabels) or @srsName">Присутність атрибуту axisLabels
означає присутність атрибуту srsName.</sch:report>
    <sch:report test="not(@uomLabels) or @srsName">Присутність атрибуту uomLabels
означає присутність атрибуту srsName attribute.</sch:report>
    <sch:report test="(not(@uomLabels) and not(@axisLabels)) or (@uomLabels and
@axisLabels)">Присутність uomLabels означає присутність атрибуту of the axisLabels і
навпаки.</sch:report>
  </sch:rule>
```

</sch:pattern>

Екземпляри прямих позицій зберігають координати позиції в деякій референційній системі координат (РСК). Оскільки прямі позиції, як типи даних, будуть часто включені в більші об'єкти (такі, як елементи геометрії), що прив'язані до РСК, атрибут `srsName` у загальному випадку буде відсутній, якщо ця специфічна пряма позиція включена до більшого об'єкту з такою ж прив'язкою до референційної системи координат. У такому разі цю систему неявно вважають за таку, що приймає значення референційної системи координат об'єкту, що міститься в більшому об'єкті.

Група атрибутів `SRSReferenceGroup` описана в підпункті 10.1.3.2. Якщо атрибут `srsName` не поданий, то РСК треба визначити як частину більшого контексту, частиною якого є цей елемент геометрії, звичайно геометричний об'єкт типу точки, лінії і т. і.

**Примітка 1.** Очікується, що атрибут лише іноді буде визначений на рівні прямої позиції.

У додатку Г.2.3.4 визначено реалізацію для `DirectPosition` (ISO 19107) з допомогою цих компонентів схеми.

#### 10.1.4.2 `DirectPositionListType`, `posList`

```
<complexType name="DirectPositionListType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
      <attribute name="count" type="positiveInteger" />
    </extension>
  </simpleContent>
</complexType>

<element name="posList" type="gml:DirectPositionListType" />
```

Екземпляри `posList` (та інші екземпляри з моделями контенту, визначені через `DirectPositionListType`) містять координати для послідовності прямих позицій у тій же самій референційній системі координат (РСК).

Група атрибутів “`SRSReferenceGroup`” описана в підпункті 10.1.3.2. Якщо не подано жодного атрибуту для `srsName`, то референційна система координат має бути визначена як частина ширшого контексту, частиною якого є цей елемент геометрії, звичайно геометричний об'єкт типу точки, кривої і т. і.

**Примітка 1.** Очікується, що атрибут `srsName` лише іноді буде визначений на рівні прямої позиції.

Необов'язковий атрибут `count` визначає число прямих позицій у списку. Якщо атрибут `count` присутній, то атрибут `srsDimension` має також бути присутнім.

Довжина списку дорівнює добутку розмірності референційної системи координат (тобто, отриманого значення з визначення референційної системи координат) та числа прямих позицій.

У додатку Г.2.3.4 визначене реалізацію геометричного об'єкту `GM_PointArray` (ISO 19107) з використанням прямих позицій тільки для цих компонентів схеми.

#### 10.1.4.3 `geometricPositionGroup`

```
<group name="geometricPositionGroup">
  <choice>
    <element ref="gml:pos"/>
```

```

    <element ref="gml:pointProperty"/>
  </choice>
</group>

```

GML підтримує два різних способи визначення геометричної позиції: або через пряму позицію (тип даних), або через точку (геометричний об'єкт).

Елементи gml:pos є позиціями, що “належать” геометричному примітиву, що інкапсулює цю геометричну позицію.

Елементи gml:pointProperty містять точку, на яку може бути спрямоване посилання з інших елементів геометрії або яка може посилатися на іншу точку, визначену десь іще (повторне використання існуючих точок).

У додатку Г.2.3.4 визначена реалізація для GM\_Position стандарту ISO 19107 з допомогою цієї групи вибору.

#### 10.1.4.4 geometricPositionListGroup

```

<group name="geometricPositionListGroup">
  <choice>
    <element ref="gml:posList"/>
    <group ref="gml:geometricPositionGroup" maxOccurs="unbounded"/>
  </choice>
</group>

```

GML підтримує два різних шляхи визначення списку геометричних позицій: або послідовністю геометричних позицій (через повторне використання визначення групи), або послідовністю прямих позицій (елемент gml:posList).

Елемент gml:posList забезпечує компактний спосіб визначення координати позицій, якщо всі позиції подані в одній системі координат.

У додатку Г.2.3.4 визначено реалізацію геометричного об'єкту GM\_PointArray (ISO 19107) через цю групу вибору.

**Примітка 1.** У визначенні цієї групи можна вживати як шаблон визначення геометричних примітивів замість використання цього визначення групи безпосередньо. Головною зміною звичайно буде зміна чисельності групи, на яку спрямоване посилання. LineString, наприклад, потребує хоча б дві позиції.

Також, для підтримки скасованих елементів, наприклад, gml:coordinates (замінений на gml:posList) та gml:pointRep (замінений на gml:pointProperty) у поточних кодах масиву точок у GML, тобто, у криволінійних відрізках, цю групу використовують як шаблон та додають скасовані елементи.

#### 10.1.4.5 VectorType, vector

```

<complexType name="VectorType">
  <simpleContent>
    <restriction base="gml:DirectPositionType"/>
  </simpleContent>
</complexType>

<element name="vector" type="gml:VectorType" />

```

gml:vector реалізує ISO/TS 19103 Vector (див. додаток Г.2.3.2 та ISO/TS 19103:2005, підпункт 6.5.2.6).



Для деяких аплікацій компоненти позиції можна пристосувати для отримання одиничного вектора.

**Примітка 1.** Це визначення допускає використання VectorType всюди, де доречно – наприклад, для offsetVector в grids.xsd, та пряме використання вектору всюди, де доречно – наприклад, у DirectionVector у direction.xsd.

#### 10.1.4.6 EnvelopeType, Envelope

```
<complexType name="EnvelopeType">
  <choice>
    <sequence>
      <element name="lowerCorner" type="gml:DirectPositionType"/>
      <element name="upperCorner" type="gml:DirectPositionType"/>
    </sequence>
    <element ref="gml:pos" minOccurs="2" maxOccurs="2"/>
    <element ref="gml:coordinates"/>
  </choice>
  <attributeGroup ref="gml:SRSReferenceGroup"/>
</complexType>
```

```
<element name="Envelope" type="gml:EnvelopeType"
substitutionGroup="gml:AbstractObject"/>
```

gml:Envelope реалізує геометричний об'єкт GM\_Envelope з ISO 19107 (див. додаток Г.2.3.4 та ISO 19107:2003, 6.4.3).

Обмежувальний простір визначає екстент із допомогою пари позицій, що визначають протилежні кути у довільних вимірах. Перша пряма позиція є "нижнім кутом" (координатною позицією, що складається з усіх мінімальних ординат по кожному виміру для всіх точок всередині обмежувального простору), а друга - "верхнім кутом" (координатною позицією, що складається з усіх максимальних координат по кожному виміру для всіх точок всередині обмежувального простору).

*Використання властивостей "coordinates" та "pos" скасовано. Натомість треба використовувати явно іменовані властивості "lowerCorner" and "upperCorner" (нижній та верхній кути).*

**Примітка 1.** Незалежно від розмірності, обмежувальний простір можна подавати однозначно як дві прямі позиції (координатні точки), якщо порядок цих точок відповідає визначеному правилу. Однак, обмежувальний простір не завжди визначає мінімальний прямокутний обмежувальний регіон, якщо референсна система, на яку спрямоване посилання, є геодезичною, або використана еліпсоїдна, сферична, полярна або циліндрична система координат, як ці терміни визначені в підрозділі 12.4. Зокрема, цей простір не може визначати мінімальний обмежувальний прямокутний регіон геометрії, множина точок якого охоплює небезперервність значень в кутовій координатній осі. Такі вісі включають довготу та широту еліпсоїдної та сферичної систем координат. Така геометрія може лежати в межах малого регіону на поверхні еліпсоїда чи сфери, або може покривати всю сферу чи весь еліпсоїд.

## 10.2 Абстрактні геометричні примітиви

### 10.2.1 AbstractGeometricPrimitiveType, AbstractGeometricPrimitive

```
<complexType name="AbstractGeometricPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometryType" />
  </complexContent>
</complexType>
```

```
</complexContent>
</complexType>
```

```
<element name="AbstractGeometricPrimitive" type="gml:AbstractGeometricPrimitiveType" abstract="true"
  substitutionGroup="gml:AbstractGeometry" />
```

`gml:AbstractGeometricPrimitiveType` є абстрактним кореневим типом геометричних примітивів. Геометричний примітив є геометричним об'єктом, який не можна розкласти далі на інші примітиви в системі. Всі примітиви орієнтовані в напрямку, закладеному в послідовності кортежів їх координат.

Елемент `gml:AbstractGeometricPrimitive` є абстрактним головою у групі заміни для всіх (наперед визначених та визначених користувачем) геометричних примітивів.

`gml:AbstractGeometricPrimitive` реалізує геометричний примітив із стандарту ISO 19107 GM\_Primitive (див. Г.2.3.3 та ISO 19107:2003, пункт 6.3.10).

### 10.2.2 GeometricPrimitivePropertyType

```
<complexType name="GeometricPrimitivePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeometricPrimitive" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

Властивість, доменом значень якої є геометричний примітив, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або посиланням XLink на віддалений елемент геометрії (де віддалений означає розміщений десь іще в тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

## 10.3 Геометричні примітиви (нульвимірні)

### 10.3.1 PointType, Point

```
<complexType name="PointType">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType">
      <sequence>
        <choice>
          <element ref="gml:pos" />
          <element ref="gml:coordinates" />
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Point" type="gml:PointType"
  substitutionGroup="gml:AbstractGeometricPrimitive" />
```

Точку `gml:Point` визначають через один кортеж координат. Пряма позиція точки визначена через елемент `gml:pos` типу `gml:DirectPositionType`.

`gml:Point` реалізує геометричний об'єкт "точка" `GM_Point` із стандарту ISO 19107 (див. Г.2.3.3 та ISO 19107:2003, пункт 6.3.11).

Використання елемента "`coordinates`" скасоване; натомість треба вживати "`pos`".

### 10.3.2 PointPropertyType, pointProperty

```
<complexType name="PointPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Point"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="pointProperty" type="gml:PointPropertyType" />
```

Властивість, значенням домену якої є точка, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалений означає елемент геометрії, розміщений десь ще у тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

Цей елемент властивості або посилається на точку через XLink атрибути, або містить точковий елемент. `pointProperty` є наперед визначеною властивістю, що можна вживати в прикладній GML схемі завжди, коли GML географічний об'єкт має властивість, яка може замінити `Point`.

### 10.3.3 PointArrayPropertyType, pointArrayProperty

```
<complexType name="PointArrayPropertyType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:Point" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="pointArrayProperty" type="gml:PointArrayPropertyType" />
```

`gml:PointArrayPropertyType` є вмістилищем для будь-якого масиву точок. Елементи завжди містяться як включені у властивість масиву; посилання на елементи геометрії або масиву елементів геометрії через XLink посилання не підтримуються.

Цей елемент властивості містить список точкових елементів. `pointArrayProperty` є наперед визначеною властивістю, яку можна використовувати у GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яке може замінити список точок.

## 10.4 Геометричні примітиви (одновимірні)

### 10.4.1 AbstractCurveType, AbstractCurve

```
<complexType name="AbstractCurveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType"/>
  </complexContent>
</complexType>
```

```
</complexContent>
</complexType>
```

```
<element name="AbstractCurve" type="gml:AbstractCurveType" abstract="true"
  substitutionGroup="gml:AbstractGeometricPrimitive" />
```

`gml:AbstractCurveType` є абстракцією кривої для підтримки різних рівнів складності. Криву завжди можна розглядати як геометричний примітив, тобто як безперервну.

Елемент `AbstractCurve` є абстрактним головою групи заміни для всіх (безперервних) елементів кривої.

#### 10.4.2 CurvePropertyType, curveProperty

```
<complexType name="CurvePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractCurve"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

```
<element name="curveProperty" type="gml:CurvePropertyType" />
```

Властивість, значенням домену якої є крива, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалений означає геометричні елементи, розміщені десь ще у тому ж самому документі). Треба обов'язково подавати тільки елемент, що міститься, або тільки посилання на нього.

Цей елемент властивості або зв'язаний посиланням з кривою через XLink атрибуту, або містить елемент кривої. `curveProperty` є наперед визначеною властивістю, що можна використовувати в GML прикладних схемах завжди, коли GML географічний об'єкт має властивість, яка може замінити `AbstractCurve`.

#### 10.4.3 CurveArrayPropertyType, curveArrayProperty

```
<complexType name="CurveArrayPropertyType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractCurve" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

```
<element name="curveArrayProperty" type="gml:CurveArrayPropertyType" />
```

Є вмістилищем для масиву кривих. Елементи завжди містяться як включені у властивості масиву; XLink посилання на геометричні об'єкти не підтримуються. Цей елемент властивості містить список елементів кривої. `curveArrayProperty` є наперед визначеною властивістю, яку можна використовувати у GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яке може замінити список кривих.

#### 10.4.4 LineStringType, LineString

```

<complexType name="LineStringType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <choice>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
            <element ref="gml:pointRep"/>
          </choice>
          <element ref="gml:posList"/>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<element name="LineString" type="gml:LineStringType"
substitutionGroup="gml:AbstractCurve" />

```

`gml:LineString` є особливою лінією, що складається з простого відрізка з лінійною інтерполяцією (див. Г.3.5). Вона визначена двома або більше кортежами координатними, з лінійною інтерполяцією між ними.

Кодування контрольних точок відповідає шаблону, описаному в підпункті 10.1.4.4. Число прямих позицій у списку має бути не менше двох.

**Примітка 1.** Геометричний об'єкт `GM_LineString` (ISO 19107) реалізований з допомогою `gml:LineStringSegment`.

#### 10.4.5 CurveType, Curve

```

<complexType name="CurveType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <element ref="gml:segments" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<element name="Curve" type="gml:CurveType" substitutionGroup="gml:AbstractCurve" />

```

`gml:Curve` реалізує `GM_Curve` стандарту ISO 19107 (див. Г.2.3.3 та ISO 19107:2003, пункт 6.3.16).

Крива є одновимірним примітивом. Криві є безперервними, зв'язними та мають довжину, яку можна виміряти в контексті системи координат.

Крива складається з одного чи більше криволінійних відрізків. Кожний такий відрізок у складі кривої треба визначити з допомогою окремого методу інтерполяції.

Відрізки кривої зв'язані один із іншим, так, що остання точка кожного відрізка, за виключенням останнього, є першою точкою іншого відрізка у списку відрізків. Орієнтація кривої є додатною.

Елемент `gml:segments` інкапсулює відрізки кривої.

#### 10.4.6 `OrientableCurveType`, `OrientableCurve`, `baseCurve`

```
<complexType name="OrientableCurveType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <element ref="gml:baseCurve" />
      </sequence>
      <attribute name="orientation" type="gml:SignType" default="+" />
    </extension>
  </complexContent>
</complexType>
```

```
<element name="baseCurve" type="gml:CurvePropertyType" />
```

```
<element name="OrientableCurve" type="gml:OrientableCurveType"
substitutionGroup="gml:AbstractCurve" />
```

`gml:OrientableCurve` реалізує геометричний об'єкт `GM_OrientableCurve`, визначений в ISO 19107 (див. Г.2.3.3 та ISO 19107:2003, підпункт 6.3.14).

`gml:OrientableCurve` складається з кривої та орієнтації. Якщо орієнтація `orientation` додатна, то `gml:OrientableCurve` є ідентичною до `gml:baseCurve`. Якщо орієнтація `orientation` від'ємна, то `gml:OrientableCurve` пов'язана з іншою абстрактною кривою `gml:AbstractCurve` із параметризацією, що обертає смисл криволінійної трансверсалі.

Властивість `gml:baseCurve` має містити базисну криву або посилається на неї, тобто або зв'язана посиланням із базовою кривою через `XLink` атрибути, або містить елемент кривої. Елемент кривої – це будь-який елемент, яким можна замінити `AbstractCurve`. Базова крива має додатну орієнтацію.

**Примітка 1.** Це визначення допускає вкладені структури, тобто `gml:OrientableCurve` може мати іншу `gml:OrientableCurve` як свою базову криву.

#### 10.4.7 Криволінійні відрізки

##### 10.4.7.1 `AbstractCurveSegmentType`, `AbstractCurveSegment`

```
<complexType name="AbstractCurveSegmentType" abstract="true">
  <attribute name="numDerivativesAtStart" type="integer" default="0" />
  <attribute name="numDerivativesAtEnd" type="integer" default="0" />
  <attribute name="numDerivativeInterior" type="integer" default="0" />
</complexType>
```

```
<element name="AbstractCurveSegment" type="gml:AbstractCurveSegmentType" abstract="true"
substitutionGroup="gml:AbstractObject" />
```

`gml:AbstractCurveSegment` реалізує геометричний об'єкт `GM_CurveSegment`, визначений в ISO 19107 (див. Г.2.3.3 та ISO 19107:2003, пункт 6.4.9).

Відрізок кривої визначає однорідний відрізок кривої.

Атрибути numDerivativesAtStart, numDerivativesAtEnd та numDerivativesInterior визначають тип та безперервність, як це визначено в ISO 19107:2003, підпункт 6.4.9.3.

Елемент gml:AbstractCurveSegment є абстрактним головою в групі заміни для всіх елементів відрізка кривої, тобто безперервних відрізків, побудованих через один і той же механізм інтерполяції.

Кодування контрольних точок у відрізка кривої має відповідати шаблону, описаному в підпункті 10.1.4.4.

Всі відрізки кривої повинні мати атрибут interpolation з типом інтерполяції gml:CurveInterpolationType, що визначає механізм криволінійної інтерполяції для даного відрізка. У цьому механізмі використовують контрольні точки та контрольні параметри, що визначають позицію цього відрізка кривої.

#### 10.4.7.2 CurveSegmentArrayPropertyType, segments

```
<complexType name="CurveSegmentArrayPropertyType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractCurveSegment" />
  </sequence>
</complexType>
```

gml:CurveSegmentArrayPropertyType є вмістилищем для масиву відрізків кривої.

```
<element name="segments" type="gml:CurveSegmentArrayPropertyType" />
```

Цей елемент властивості містить список відрізків кривої. Порядок елементів має значення та має бути збереженим у ході обробки масиву.

#### 10.4.7.3 CurveInterpolationType

```
<simpleType name="CurveInterpolationType">
  <restriction base="string">
    <enumeration value="linear" />
    <enumeration value="geodesic" />
    <enumeration value="circularArc3Points" />
    <enumeration value="circularArc2PointWithBulge" />
    <enumeration value="circularArcCenterPointWithRadius" />
    <enumeration value="elliptical" />
    <enumeration value="clothoid" />
    <enumeration value="conic" />
    <enumeration value="polynomialSpline" />
    <enumeration value="cubicSpline" />
    <enumeration value="rationalSpline" />
  </restriction>
</simpleType>
```

gml:CurveInterpolationType є списком кодів, які треба використовувати для ідентифікації механізмів інтерполяції, визначених у прикладній схемі.

Цей тип реалізує геометричний об'єкт GM\_CurveInterpolation, визначений у ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.8).

#### 10.4.7.3 LineStringSegmentType, LineStringSegment

```
<complexType name="LineStringSegmentType">
```

```

<complexContent>
  <extension base="gml:AbstractCurveSegmentType">
    <sequence>
      <choice>
        <choice minOccurs="2" maxOccurs="unbounded">
          <element ref="gml:pos" />
          <element ref="gml:pointProperty" />
          <element ref="gml:pointRep"/>
        </choice>
        <element ref="gml:posList" />
        <element ref="gml:coordinates" />
      </choice>
    </sequence>
    <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="linear" />
  </extension>
</complexContent>
</complexType>

```

```

<element name="LineStringSegment" type="gml:LineStringSegmentType"
  substitutionGroup="gml:AbstractCurveSegment" />

```

`gml:LineStringSegment` реалізує геометричний об'єкт `GM_LineString`, визначений в ISO 19107 (див. Г.2.3.4 and ISO 19107:2003, пункт 6.4.10).

`LineStringSegment` є відрізком кривої, визначеним двома чи більше контрольними точками, включно із першою та останньою точками, з лінійною інтерполяцією між ними.

Модель контенту відповідає загальному шаблону для кодування відрізків кривих (див. пункт 10.4.7).

#### 10.4.7.5 `ArcStringType`, `ArcString`

```

<complexType name="ArcStringType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc3Points" />
      <attribute name="numArc" type="integer" />
    </extension>
  </complexContent>

```



```
</complexType>
```

```
<element name="ArcString" type="gml:ArcStringType"
substitutionGroup="gml:AbstractCurveSegment" />
```

`gml:ArcString` реалізує геометричний об'єкт `GM_ArcString`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.14).

`ArcString` є відрізком кривої, в якому використовують круглу дугову інтерполяцію по трьох точках (“`circularArc3Points`”). Число дуг в цьому відрізку можна явно вказати в атрибуті `numArc`. Число контрольних точок у цьому відрізку має становити  $2 * \text{numArc} + 1$ .

Модель контенту відповідає загальному шаблону для кодування відрізків кривої (див. пункт 10.4.7).

#### 10.4.7.6 ArcType, Arc

```
<complexType name="ArcType">
  <complexContent>
    <restriction base="gml:ArcStringType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
      </sequence>
      <attribute name="numArc" type="integer" fixed="1" />
    </restriction>
  </complexContent>
</complexType>
```

```
<element name="Arc" type="gml:ArcType" substitutionGroup="gml:ArcString" />
```

`gml:Arc` реалізує геометричний об'єкт `GM_Arc`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.15).

`Arc` є дугою, що складається тільки з однієї одиниці дуги, тобто з трьох контрольних точок, які включають першу та останню точки; тому атрибут “`numArc`” фіксовано дорівнює “1”.

#### 10.4.7.7 CircleType, Circle

```
<complexType name="CircleType">
  <complexContent>
    <extension base="gml:ArcType" />
  </complexContent>
</complexType>
```

```
<element name="Circle" type="gml:CircleType" substitutionGroup="gml:Arc" />
```

`gml:Circle` впроваджує геометричний об'єкт коло `GM_Circle`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.16).

Коло є дугою, кінці якої збігаються, утворюючи просту замкнену петлю. Три контрольні точки мають бути різними неколінеарними точками для кола з тим, щоб уникнути їх неоднозначного визначення. Така дуга просто продовжується за третю контрольну точку, доки не досягне першої контрольної точки.

#### 10.4.7.8 `ArcStringByBulgeType`, `ArcStringByBulge`

```
<complexType name="ArcStringByBulgeType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <choice>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
        <element name="bulge" type="double" maxOccurs="unbounded"/>
        <element name="normal" type="gml:VectorType" maxOccurs="unbounded" />
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
        fixed="circularArc2PointWithBulge" />
      <attribute name="numArc" type="integer" />
    </extension>
  </complexContent>
</complexType>

<element name="ArcStringByBulge" type="gml:ArcStringByBulgeType"
  substitutionGroup="gml:AbstractCurveSegment" />
```

`gml:ArcStringByBuldge` реалізує геометричний об'єкт `GM_ArcStringByBuldge`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.17).

За цим варіантом дуги обчислюють середні точки дуги замість прямого зберігання координат. Послідовність контрольних точок складається з першої та останньої точки кожної дуги плюс `gml:bulge` (див. ISO 19107:2003, підпункт 6.4.17.2). `gml:normal` є перпендикуляром до хорди дуги (див. ISO 19107:2003, підпункт 6.4.17.4).

`interpolation` фіксують як `"circularArc2PointWithBulge"`.

Число дуг в відрізку дуги треба явно вказати в атрибуті `numArc`. Число контрольних точок у відрізку дуги має становити `numArc` плюс 1.

Модель контенту відповідає загальному зразку для кодування відрізків кривої (див. підпункт 11.2.1.3).

#### 10.4.7.9 `ArcByBulgeType`, `ArcByBulge`

```
<complexType name="ArcByBulgeType">
  <complexContent>
```

```

<restriction base="gml:ArcStringByBulgeType">
  <sequence>
    <choice>
      <choice minOccurs="2" maxOccurs="2">
        <element ref="gml:pos" />
        <element ref="gml:pointProperty" />
        <element ref="gml:pointRep" />
      </choice>
      <element ref="gml:posList" />
      <element ref="gml:coordinates" />
    </choice>
    <element name="bulge" type="double" />
    <element name="normal" type="gml:VectorType"/>
  </sequence>
  <attribute name="numArc" type="integer" fixed="1" />
</restriction>
</complexContent>
</complexType>

```

```

<element name="ArcByBulge" type="gml:ArcByBulgeType"
substitutionGroup="gml:ArcStringByBulge" />

```

gml:ArcByBulge реалізує геометричний об'єкт GM\_ArcByBulge, визначений в ISO 19107 (див. Г.2.3.4 and ISO 19107:2003, пункт 6.4.18).

ArcByBulge є відрізком дуги тільки з однією дуговою одиницею, тобто дві контрольні точки, один bulge та один перпендикулярний вектор.

Дуга є відрізком дуги, що складається з однієї дуги; атрибут "numArc" дорівнює 1.

#### 10.4.7.10 ArcByCenterPointType, ArcByCenterPoint

```

<complexType name="ArcByCenterPointType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <choice>
          <choice>
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
        <element name="radius" type="gml:LengthType" />
        <element name="startAngle" type="gml:AngleType" minOccurs="0" />
        <element name="endAngle" type="gml:AngleType" minOccurs="0" />
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
        fixed="circularArcCenterPointWithRadius" />
      <attribute name="numArc" type="integer" use="required" fixed="1" />
    </extension>
  </complexContent>

```

```
</complexContent>
</complexType>
```

```
<element name="ArcByCenterPoint" type="gml:ArcByCenterPointType"
  substitutionGroup="gml:AbstractCurveSegment" />
```

У цьому варіанті дуги точки дуги треба обчислити замість прямого зберігання координат. Одиначною контрольною точкою є центральна точка дуги, доповнена радіусом та напрямком на початку та в кінці.

Це подання можна використати лише у двохвимірній системі.

Елемент `gml:radius` визначає радіус дуги.

Елемент `gml:startAngle` визначає напрямок дуги в її початку.

Елемент `gml:endAngle` визначає напрямок дуги в її кінці.

`interpolation` фіксовано як `"circularArcCenterPointWithRadius"`.

Оскільки цей тип описує лише одиначну дугу, атрибут `"numArc"` становить 1.

Модель контенту відповідає загальному шаблону для кодування відрізків кривої (див. пункт 10.4.7).

#### 10.4.7.11 CircleByCenterPointType, CircleByCenterPoint

```
<complexType name="CircleByCenterPointType">
  <complexContent>
    <restriction base="gml:ArcByCenterPointType">
      <sequence>
        <choice>
          <choice>
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
        <element name="radius" type="gml:LengthType" />
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
<element name="CircleByCenterPoint" type="gml:CircleByCenterPointType"
  substitutionGroup="gml:ArcByCenterPoint" />
```

`gml:CircleByCenterPoint` є `gml:ArcByCenterPoint` з ідентичним початковим та кінцевим кутом для формування повного кола. Це подання також можна використати лише у двохвимірній системі.

#### 10.4.7.12 CubicSplineType, CubicSpline

```
<complexType name="CubicSplineType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
```

```

<choice>
  <choice minOccurs="2" maxOccurs="unbounded">
    <element ref="gml:pos" />
    <element ref="gml:pointProperty" />
    <element ref="gml:pointRep" />
  </choice>
  <choice>
    <element ref="gml:posList" />
    <element ref="gml:coordinates" />
  </choice>
  <element name="vectorAtStart" type="gml:VectorType" />
  <element name="vectorAtEnd" type="gml:VectorType" />
</sequence>
<attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="cubicSpline" />
<attribute name="degree" type="integer" fixed="3" />
</extension>
</complexContent>
</complexType>

<element name="CubicSpline" type="gml:CubicSplineType"
substitutionGroup="gml:AbstractCurveSegment" />

```

`gml:CubicSpline` реалізує геометричний об'єкт кубічний сплайн `GM_CubicSpline`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.28).

Число контрольних точок має бути не менше трьох.

`gml:vectorAtStart` – це одиничний вектор, що торкається сплайну у початковій точці. `vectorAtEnd` – це одиничний вектор, що торкається сплайну у кінцевій точці. Для визначення форми кубічного сплайну треба використовувати тільки напрямки векторів, а не їх довжину.

`interpolation` дорівнює значенню "cubicSpline".

`degree` має бути ступенем поліному, використаного для інтерполяції в цьому сплайні. Тому ступінь для кубічного сплайну становить 3.

Модель контенту відповідає загальному шаблону для кодування відрізків кривої (див. пункт 10.4.7).

#### 10.4.7.13 BSplineType, BSpline

```

<complexType name="BSplineType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <choice>
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

    <element name="degree" type="nonNegativeInteger" />
    <element name="knot" type="gml:KnotPropertyType" minOccurs="2"
      maxOccurs="unbounded" />
  </sequence>
  <attribute name="interpolation" type="gml:CurveInterpolationType"
default="polynomialSpline" />
  <attribute name="isPolynomial" type="boolean" />
  <attribute name="knotType" type="gml:KnotTypesType" />
</extension>
</complexContent>
</complexType>

```

```

<element name="BSpline" type="gml:BSplineType"
substitutionGroup="gml:AbstractCurveSegment" />

```

`gml:BSpline` реалізує геометричний об'єкт В-сплайн `GM_BSplineCurve`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.30).

В-Spline – це поділена на частини параметрична поліноміальна або раціональна крива, описана в термінах контрольних точок та базових функцій, визначених в ISO 19107:2003, пункт 6.4.30. Тому `interpolation` може бути або "polynomialSpline", або "rationalSpline" залежно від типу інтерполяції; значенням за умовчанням є "polynomialSpline".

`degree` має бути ступенем поліному, застосованого для інтерполяції цього сплайну. `gml:knot` має бути послідовністю окремих вузлів, використаною для визначення базових сплайнових функцій (див. ISO 19107:2003, підпункт 6.4.26.2).

Атрибут `isPolynomial` має дорівнювати "true" ("істина"), якщо це є поліноміальний сплайн (див. ISO 19107:2003, підпункт 6.4.30.5).

Атрибутом `knotType` треба подавати тип розподілу вузлів, який використовують у визначенні цього сплайну (див. ISO 19107:2003, підпункт 6.4.30.4).

Модель контенту має відповідати загальному шаблону для кодування відрізків кривої (див. пункт 10.4.7).

#### 10.4.7.14 KnotType, KnotPropertyType

```

<complexType name="KnotType">
  <sequence>
    <element name="value" type="double" />
    <element name="multiplicity" type="nonNegativeInteger" />
    <element name="weight" type="double" />
  </sequence>
</complexType>

```

`gml:Knot` реалізує геометричний об'єкт ISO 19107 `GM_Knot` (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.24).

Вузол є точкою перелому на сплайновій кривій, що складається з відрізків. `gml:value` є значенням параметру в вузлі сплайну (див. ISO 19107:2003, підпункт 6.4.24.2).

`gml:multiplicity` – це множинність цього вузлу, використана в визначенні сплайну (з такою ж вагою).

`gml:weight` є значенням середньої ваги, використаної для цього вузлу сплайну.

```

<complexType name="KnotPropertyType">
  <sequence>
    <element name="Knot" type="gml:KnotType" />
  </sequence>
</complexType>

```

`gml:KnotPropertyType` інкапсулює вузол для використання в геометричному типі.

#### 10.4.7.15 KnotTypesType

```

<simpleType name="KnotTypesType">
  <restriction base="string">
    <enumeration value="uniform" />
    <enumeration value="quasiUniform" />
    <enumeration value="piecewiseBezier" />
  </restriction>
</simpleType>

```

`gml:KnotTypesType` реалізує геометричний об'єкт `knot GM_KnotType`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.25).

Цей тип переліку визначає значення для типу вузлу (див. ISO 19107:2003, пункт 6.4.25).

#### 10.4.7.16 BezierType, Bezier

```

<complexType name="BezierType">
  <complexContent>
    <restriction base="gml:BSplineType">
      <sequence>
        <choice>
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:posList" />
          <element ref="gml:coordinates" />
        </choice>
        <element name="degree" type="nonNegativeInteger" />
        <element name="knot" type="gml:KnotPropertyType" minOccurs="2"
maxOccurs="2" />
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="polynomialSpline" />
      <attribute name="isPolynomial" type="boolean" fixed="true" />
      <attribute name="knotType" type="gml:KnotTypesType" use="prohibited" />
    </restriction>
  </complexContent>
</complexType>

<element name="Bezier" type="gml:BezierType" substitutionGroup="gml:BSpline" />

```

`gml:Bezier` реалізує геометричний об'єкт `GM_Bezier`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.31).

Криві Безьє – це поліноміальні сплайни, в яких використаний поліном Безьє або Бернштейна для цілей інтерполяції. Це є особливий випадок B-сплайну з двома вузлами.

`gml:degree` має бути ступенем поліному, використаного для інтерполяції цього сплайну.

`gml:knot` має бути послідовністю окремих вузлів для визначення сплайнових базисних функцій.

`interpolation` дорівнює як `"polynomialSpline"`.

`isPolynomial` дорівнює `"true"` ("істина").

`knotType` є недоречним для відрізків кривої Безьє.

#### **10.4.7.17 OffsetCurveType, OffsetCurve**

```
<complexType name="OffsetCurveType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <element name="offsetBase" type="gml:CurvePropertyType"/>
        <element name="distance" type="gml:LengthType"/>
        <element name="refDirection" type="gml:VectorType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="OffsetCurve" type="gml:OffsetCurveType"
substitutionGroup="gml:AbstractCurveSegment"/>
```

Зміщена крива (`OffsetCurve`) є кривою на певній віддалі від базисної кривої.

`gml:OffsetCurve` реалізує геометричний об'єкт `GM_OffsetCurve` стандарту ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.23).

`gml:offsetBase` є базовою кривою, від якої ця крива визначена як зміщена крива. `gml:distance` та `gml:refDirection` мають той же смисл, що і в ISO 19107:2003, пункт 6.4.23.

Модель контенту відповідає загальному шаблону для кодування відрізків кривої (див. підпункт 10.4.7).

#### **10.4.7.18 AffinePlacementType, AffinePlacement**

```
<complexType name="AffinePlacementType">
  <sequence>
    <element name="location" type="gml:DirectPositionType"/>
    <element name="refDirection" type="gml:VectorType" maxOccurs="unbounded"/>
    <element name="inDimension" type="positiveInteger"/>
    <element name="outDimension" type="positiveInteger"/>
  </sequence>
</complexType>
```

```
<element name="AffinePlacement" type="gml:AffinePlacementType"
substitutionGroup="gml:AbstractObject"/>
```



`gml:AffinePlacement` реалізує геометричний об'єкт афінне розміщення `GM_AffinePlacement`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.21 та підпункт 6.4.20.1). `gml:location`, `gml:refDirection`, `gml:inDimension` та `gml:outDimension` мають той же смисл, що і в ISO 19107:2003, пункт 6.4.21.

#### 10.4.7.19 ClothoidType, Clothoid

```

<complexType name="ClothoidType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <element name="refLocation">
          <complexType>
            <sequence>
              <element ref="gml:AffinePlacement"/>
            </sequence>
          </complexType>
        </element>
        <element name="scaleFactor" type="decimal"/>
        <element name="startParameter" type="double"/>
        <element name="endParameter" type="double"/>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="clothoid">
    </extension>
  </complexContent>
</complexType>

<element name="Clothoid" type="gml:ClothoidType"
substitutionGroup="gml:AbstractCurveSegment"/>

```

Клотоїд, або спіраль Корну, є плоскою кривою, кривизна якої є фіксованою функцією її довжини.

`gml:Clothoid` реалізує геометричний елемент `GM_Clothoid`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.22).

`gml:refLocation`, `gml:startParameter`, `gml:endParameter` та `gml:scaleFactor` мають той же смисл, що і в ISO 19107:2003, пункт 6.4.22.

`interpolation` дорівнює "clothoid".

Модель контенту відповідає загальному шаблону для кодування відрізків кривої (див. підпункт 10.4.7).

#### 10.4.7.20 GeodesicStringType, GeodesicString

```

<complexType name="GeodesicStringType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <choice>
        <element ref="gml:posList"/>
        <group ref="gml:geometricPositionGroup" minOccurs="2"
maxOccurs="unbounded"/>
      </choice>

```

```

        <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="geodesic"/>
    </extension>
</complexContent>
</complexType>

```

```

<element name="GeodesicString" type="gml:GeodesicStringType"
substitutionGroup="gml:AbstractCurveSegment"/>

```

gml:GeodesicString реалізує геометричний об'єкт GM\_GeodesicString, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.12), що є послідовністю геодезичних відрізків.

Число контрольних точок має бути не менше двох.

interpolation дорівнює "geodesic".

Модель контенту відповідає загальному шаблону кодування для відрізків кривої (див. підпункт 10.4.7).

#### **10.4.7.21 GeodesicType, Geodesic**

```

<complexType name="GeodesicType">
    <complexContent>
        <extension base="gml:GeodesicStringType"/>
    </complexContent>
</complexType>

```

```

<element name="Geodesic" type="gml:GeodesicType"
substitutionGroup="gml:GeodesicString"/>

```

gml:Geodesic реалізує геометричний об'єкт GM\_Geodesic, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, підпункт 6.4.13).

### **10.5 Геометричні примітиви (двохвимірні)**

#### **10.5.1 AbstractSurfaceType, AbstractSurface**

```

<complexType name="AbstractSurfaceType" abstract="true">
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType"/>
    </complexContent>
</complexType>

```

```

<element name="AbstractSurface" type="gml:AbstractSurfaceType" abstract="true"
substitutionGroup="gml:AbstractGeometricPrimitive" />

```

gml:AbstractSurfaceType є абстракцією поверхні для підтримки різних рівнів складності. Поверхня завжди є безперервною областю площини.

Елемент gml:AbstractSurface є абстрактним головою у групі заміни для всіх (безперервних) елементів поверхні.

#### **10.5.2 SurfacePropertyType, surfaceProperty**

```

<complexType name="SurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="gml:AbstractSurface"/>
    </sequence>

```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="surfaceProperty" type="gml:SurfacePropertyType" />

```

Властивість, значенням домену якої є поверхня, може або бути відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалений означає елемент геометрії, розташований десь ще у тому ж самому документі). Треба обов'язково подавати тільки елемент, що міститься, або тільки посилання на нього.

Цей елемент властивості або посиляється на поверхню через XLink атрибути, або містить елемент поверхні.

surfaceProperty є наперед визначеною властивістю, яку можна використовувати в GML прикладній схемі завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити gml:AbstractSurface.

### 10.5.3 SurfaceArrayType, surfaceArrayProperty

```

<complexType name="SurfaceArrayType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractSurface" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

gml:SurfaceArrayType є вмістилищем для масиву поверхней. Елементи завжди містяться у властивості масиву; зв'язок елементів геометрії або масивів елементів геометрії через XLink посилання не підтримується.

```

<element name="surfaceArrayProperty" type="gml:SurfaceArrayType" />

```

Цей елемент властивості містить список елементів поверхні. surfaceArrayProperty є наперед визначеною властивістю, яку можна використати у GML прикладній схемі завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити список абстрактних поверхонь AbstractSurfaces.

### 10.5.4 PolygonType, Polygon

```

<complexType name="PolygonType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:exterior" minOccurs="0" />
        <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<element name="Polygon" type="gml:PolygonType" substitutionGroup="gml:AbstractSurface" />

```

`gml:Polygon` є особливою поверхнею, що визначена одиничною ділянкою поверхні (див. Г.3.6). Межа цієї ділянки лежить в одній площині, а у внутрішній частині полігону використана площинна інтерполяція.

**Примітка 1.** Геометричний об'єкт `GM_Polygon` (ISO 19107) реалізований з допомогою `PolygonPatch`.

Елементи `gml:exterior` та `gml:interior` описують межу поверхні полігону та визначені далі.

### 10.5.5 exterior, interior

```
<element name="exterior" type="gml:AbstractRingPropertyType" />
```

Межа поверхні складається з множини кілець. У двохвимірному випадку одне з цих кілець розрізняють як зовнішню межу. Взагалі це не завжди можливе, у випадку чого всі межі мають бути перелічені як внутрішні межі, а зовнішня має бути порожньою.

```
<element name="interior" type="gml:AbstractRingPropertyType" />
```

Межа поверхні складається з числа кілець. “Внутрішні” кільця відокремлюють поверхню чи ділянку поверхні від області, обмеженою цими кільцями.

### 10.5.6 AbstractRingType, AbstractRing

```
<complexType name="AbstractRingType" abstract="true">
  <sequence/>
</complexType>
```

```
<element name="AbstractRing" type="gml:AbstractRingType" abstract="true"
  substitutionGroup="gml:AbstractObject"/>
```

Є абстракцією кільця для підтримки меж поверхні різної складності.

Елемент `AbstractRing` є абстрактним головою у групі заміни для всіх замкнених меж ділянки поверхні.

### 10.5.7 AbstractRingPropertyType

```
<complexType name="AbstractRingPropertyType">
  <sequence>
    <element ref="gml:AbstractRing"/>
  </sequence>
</complexType>
```

Властивість із моделлю контенту `gml:AbstractRingPropertyType` інкапсулює кільце для подання властивості межі поверхні.

### 10.5.8 LinearRingType, LinearRing

```
<complexType name="LinearRingType">
  <complexContent>
    <extension base="gml:AbstractRingType">
      <sequence>
        <choice>
          <choice minOccurs="4" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointProperty" />
            <element ref="gml:pointRep" />
          </choice>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```

    </choice>
    <element ref="gml:posList" />
    <element ref="gml:coordinates" />
  </choice>
</sequence>
</extension>
</complexContent>
</complexType>

```

```
<element name="LinearRing" type="gml:LinearRingType" substitutionGroup="gml:AbstractRing" />
```

gml:LinearRing визначають чотирма чи більше кортежів координат, з лінійною інтерполяцією між ними; перші та останні координати мають збігатись.

Кодування контрольних точок відповідає шаблону, описаному в підпункті 10.1.4.4. Число прямих позицій у списку має бути не менше чотирьох.

### 10.5.9 LinearRingPropertyType

```

<complexType name="LinearRingPropertyType">
  <sequence>
    <element ref="gml:LinearRing"/>
  </sequence>
</complexType>

```

Властивість з моделлю контенту gml:LinearRingPropertyType інкапсулює лінійне кільце, яким подають компонент межі поверхні.

### 10.5.10 SurfaceType, Surface

```

<complexType name="SurfaceType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:patches" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```
<element name="Surface" type="gml:SurfaceType" substitutionGroup="gml:AbstractSurface" />
```

Поверхня є двохвимірним примітивом та складається з однієї чи більше ділянок поверхні, визначених в ISO 19107:2003, підпункт 6.3.17.1. Ділянки поверхні зв'язані одна з іншою.

gml:Surface реалізує геометричний об'єкт поверхня GM\_Surface, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.3.17).

gml:patches інкапсулює ділянки поверхні (patches).

### 10.5.11 OrientableSurfaceType, OrientableSurface, baseSurface

```

<complexType name="OrientableSurfaceType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">

```

```

<sequence>
  <element ref="gml:baseSurface" />
</sequence>
<attribute name="orientation" type="gml:SignType" default="+" />
</extension>
</complexContent>
</complexType>

```

```

<element name="baseSurface" type="gml:SurfacePropertyType" />
<element name="OrientableSurface" type="gml:OrientableSurfaceType"
substitutionGroup="gml:AbstractSurface" />

```

`gml:OrientableSurface` реалізує геометричний об'єкт `GM_OrientableSurface`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.3.15).

`gml:OrientableSurface` складається з поверхні та орієнтації. Якщо `orientation "+"`, то `gml:OrientableSurface` ідентична базовій поверхні `gml:baseSurface`. Якщо `orientation "-"`, то `gml:OrientableSurface` є посиланням на `gml:AbstractSurface` із направленою доверху нормаллю, що обертає напрямком для цієї `gml:OrientableSurface`, смисл, що вкладений у верх поверхні.

Властивість `gml:baseSurface` містить базову поверхню або посилається на неї. Властивість `gml:baseSurface` або посилається на базову поверхню через `XLink` атрибути, або містить елемент поверхні. Елементом поверхні є будь-який елемент, яким можна замінити `gml:AbstractSurface`. Базова поверхня має додатну орієнтацію.

**Примітка 1.** Це визначення допускає вкладені структури, тобто `gml:OrientableSurface` може використовувати іншу `gml:OrientableSurface` як свою базову поверхню.

#### 10.5.11.1 Ring, RingType, curveMember

```

<complexType name="RingType">
  <complexContent>
    <extension base="gml:AbstractRingType">
      <sequence>
        <element ref="gml:curveMember" maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<element name="Ring" type="gml:RingType" substitutionGroup="gml:AbstractRing" />
<element name="curveMember" type="gml:CurvePropertyType" />

```

`gml:Ring` реалізує геометричний об'єкт кільце `GM_Ring`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.3.6).

Кільце використовують для подання простого зв'язного компоненту межі поверхні, як він визначений в ISO 19107:2003, пункт 6.3.6.

Кожен `gml:curveMember` містить одну криву або посилається на неї, тобто, кожен елемент, яким можна замінити `gml:AbstractCurve`. В контексті кільця криві описують межу поверхні. В послідовності кривих треба дотримуватися суміжності та зв'язності по колу.

Якщо поданий атрибут `aggregationType`, він повинен мати значення "sequence".

**Примітка 1.** Це визначення допускає вкладену структуру, тобто `gml:curveMember` може бути композитною кривою `gml:CompositeCurve`, що, в свою чергу, має бути побудована з інших композитних кривих `gml:CompositeCurves` як членів кривої.

#### 10.5.11.2 RingPropertyType

```
<complexType name="RingPropertyType">
  <sequence>
    <element ref="gml:Ring"/>
  </sequence>
</complexType>
```

Властивість із моделлю контенту для `gml:RingPropertyType` інкапсулює кільце для подання компоненту межі поверхні.

#### 10.5.11.3 PolyhedralSurfaceType, PolyhedralSurface, polygonPatches

```
<element name="PolyhedralSurface" type="gml:SurfaceType"
substitutionGroup="gml:Surface"/>
```

`gml:PolyhedralSurface` реалізує геометричний об'єкт поліхедральна поверхня `GM_PolyhedralSurface`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.35).

Поліхедральна поверхня є поверхнею, складеною з полігональних ділянок, з'єднаних вздовж спільних межових кривих.

`gml:polygonPatches` інкапсулює полігональні ділянки поліхедральної поверхні. Всі ділянки мають бути полігональними ділянками.

#### 10.5.11.4 TriangulatedSurfaceType, TriangulatedSurface, trianglePatches

```
<element name="TriangulatedSurface" type="gml:SurfaceType"
substitutionGroup="gml:Surface"/>
```

`gml:TriangulatedSurface` реалізує геометричний об'єкт трихедральна поверхня `GM_TriangulatedSurface`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.37).

Ця поверхня є поліхедральною поверхнею, що складається лише з трикутників. Немає обмежень на спосіб отримання трикутників.

`gml:trianglePatches` інкапсулює трикутники в поверхні, що складається з трикутників. Всі ділянки мають бути трикутними ділянками.

#### 10.5.11.5 TinType, Tin

```
<complexType name="TinType">
  <complexContent>
    <extension base="gml:TriangulatedSurfaceType">
      <sequence>
        <element name="stopLines" type="gml:LineStringSegmentArrayPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="breakLines" type="gml:LineStringSegmentArrayPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="maxLength" type="gml:LengthType"/>
        <element name="controlPoint">
          <complexType>
            <choice>
```

```

        <element ref="gml:posList"/>
        <group ref="gml:geometricPositionGroup" minOccurs="3"
            maxOccurs="unbounded"/>
    </choice>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

```
<element name="Tin" type="gml:TinType" substitutionGroup="gml:TriangulatedSurface"/>
```

`gml:Tin` впроваджує геометричний об'єкт `GM_Tin`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.39).

TIN (нерегулярна мережа трикутників) є поверхнею, що складається з трикутників та в якій використаний алгоритм Делоне або подібний алгоритм, доповнений урахуванням стоп-ліній (`gml:stopLines`), структурних ліній (`gml:breakLines`), та максимальної довжини сторін трикутників (`gml:maxLength`). `gml:controlPoint` має містити множину позицій (не менше трьох) для їх використання в якості контрольних точок для TIN (кути трикутників у TIN). Див. докладніше ISO 19107:2003, пункт 6.4.39.

#### 10.5.11.6 LineStringSegmentArrayPropertyType

```

<complexType name="LineStringSegmentArrayPropertyType">
    <sequence minOccurs="0" maxOccurs="unbounded">
        <element ref="gml:LineStringSegment" />
    </sequence>
</complexType>

```

`gml:LineStringSegmentArrayPropertyType` забезпечує вмістилище для лінійних `line strings`.

#### 10.5.12 Ділянки поверхні

##### 10.5.12.1 AbstractSurfacePatchType, gml: AbstractSurfacePatch

```

<complexType name="AbstractSurfacePatchType" abstract="true" />
<element name="AbstractSurfacePatch" type="gml:AbstractSurfacePatchType" abstract="true" />

```

Ділянка поверхні визначає однорідну частину поверхні.

`gml:AbstractSurfacePatch` реалізує геометричний об'єкт ділянка поверхні `GM_SurfacePatch` стандарту ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.34).

Елемент `gml:AbstractSurfacePatch` є абстрактним головою групи заміни для всіх елементів ділянки поверхні, що описують безперервну частину поверхні.

Всі ділянки поверхні повинні мати атрибут `interpolation` (задекларований у типах, що походять від `gml:AbstractSurfacePatchType`), що визначає механізм інтерполяції для ділянки з використанням `gml:SurfaceInterpolationType` (див. підпункт 10.5.12.3).

##### 10.5.12.2 SurfacePatchArrayPropertyType, patches

```

<complexType name="SurfacePatchArrayPropertyType">
    <sequence minOccurs="0" maxOccurs="unbounded">
        <element ref="gml:AbstractSurfacePatch" />
    </sequence>

```



```

</sequence>
</complexType>

```

`gml:SurfacePatchArrayPropertyType` є вмістилищем послідовності ділянок поверхні.

```
<element name="patches" type="gml:SurfacePatchArrayPropertyType" />
```

Елемент властивості `gml:patches` містить послідовність ділянок поверхні. Порядок елементів є значущим та має зберігатися при обробці масиву.

#### 11.2.2.5 SurfaceInterpolationType

```

<simpleType name="SurfaceInterpolationType">
  <restriction base="string">
    <enumeration value="none" />
    <enumeration value="planar" />
    <enumeration value="spherical" />
    <enumeration value="elliptical" />
    <enumeration value="conic" />
    <enumeration value="tin" />
    <enumeration value="parametricCurve" />
    <enumeration value="polynomialSpline" />
    <enumeration value="rationalSpline" />
    <enumeration value="triangulatedSpline" />
  </restriction>
</simpleType>

```

`gml:SurfaceInterpolationType` є списком кодів, що можна використовувати для ідентифікації механізмів інтерполяції, визначених у прикладній схемі.

Цей тип реалізує геометричний об'єкт інтерполяція поверхні `GM_SurfaceInterpolation`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.32).

#### 10.5.12.4 PolygonPatchType, PolygonPatch

```

<complexType name="PolygonPatchType">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <element ref="gml:exterior" minOccurs="0" />
        <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
      <attribute name="interpolation" type="gml:SurfaceInterpolationType" fixed="planar" />
    </extension>
  </complexContent>
</complexType>

```

```
<element name="PolygonPatch" type="gml:PolygonPatchType"
substitutionGroup="gml:AbstractSurfacePatch" />
```

`gml:PolygonPatch` реалізує геометричний об'єкт полігон `GM_Polygon`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.36).

A `gml:PolygonPatch` є ділянкою поверхні, визначеною множиною кривих межі, та поверхнею, на якій ці криві знаходяться. Криві мають лежати в одній площині, а всередині полігону застосована площинна інтерполяція.

`gml:interpolation` дорівнює "planar", тобто інтерполяція має повертати точки, розташовані в одній і тій же площині. Межа ділянки має міститися в цій площині.

#### 10.5.12.5 TriangleType, Triangle

```
<complexType name="TriangleType">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <element ref="gml:exterior" />
      </sequence>
      <attribute name="interpolation" type="gml:SurfaceInterpolationType" fixed="planar" />
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Triangle" type="gml:TriangleType"
substitutionGroup="gml:AbstractSurfacePatch" />
```

`gml:Triangle` являє трикутник як ділянку поверхні з зовнішньою межею, що складається з лінійного кільця. Треба зауважити, що він є полігоном (підтипом) без внутрішніх меж. Число точок у лінійному кільці має становити чотири.

Кільце (елемент `gml:exterior`) має бути `gml:LinearRing` та формувати трикутник, причому перша та остання позиції мають збігатися.

`gml:interpolation` дорівнює "planar", тобто інтерполяція має повертати точки на одній і тій же площині. Межа ділянки повинна міститися в цій площині.

#### 10.5.12.6 RectangleType, Rectangle

```
<complexType name="RectangleType">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <element ref="gml:exterior" />
      </sequence>
      <attribute name="interpolation" type="gml:SurfaceInterpolationType" fixed="planar" />
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Rectangle" type="gml:RectangleType"
substitutionGroup="gml:AbstractSurfacePatch" />
```

`gml:Rectangle` являє прямокутник як ділянку поверхні з зовнішньою межею, що складається з лінійного кільця. Треба зауважити, що він є полігоном (підтип) без внутрішніх меж. Кількість точок у лінійному кільці має дорівнювати п'ять.

**Примітка 1.** Тоді як концептуально прямокутник є підтипом полігону, визначення `gml:RectangleType` як типу, отриманого через обмеження від `gml:PolygonType` є проблематичним

у зв'язку із способом, у який визначено конструкт в XML схемі, тому його уникають у цьому випадку.

Кільце (елемент `gml:exterior`) має бути `gml:LinearRing` та формувати прямокутник, причому перша та остання позиції мають співпадати.

`interpolation` дорівнює "planar", тобто, інтерполяція має повертати точки в одній і тій же площині. Межа ділянки має міститися в цій площині.

#### 10.5.12.7 PointGrid

```
<group name="PointGrid">
  <sequence>
    <element name="rows">
      <complexType>
        <sequence>
          <element name="Row" maxOccurs="unbounded">
            <complexType>
              <group ref="gml:geometricPositionListGroup"/>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
  </sequence>
</group>
```

`gml:PointGrid` впроваджує `GM_PointGrid`, визначений в ISO 19107 (дивіться ISO 19107:2003, пункт 6.4.6).

Група `gml:PointGrid` містить точки позицій, які організовані в послідовності або мережі, або посилається на них. Всі рядки `gml:row` повинні мати однакове число позицій (колонок).

#### 10.5.12.8 AbstractParametricCurveSurfaceType, AbstractParametricCurveSurface

```
<complexType name="AbstractParametricCurveSurfaceType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="AbstractParametricCurveSurface"
  type="gml:AbstractParametricCurveSurfaceType" abstract="true"
  substitutionGroup="gml:AbstractSurfacePatch"/>
```

`gml:AbstractParametricCurveSurface` реалізує геометричний об'єкт параметрична криволінійна поверхня `GM_ParametricCurveSurface`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.40).

Цей елемент надає голову групи заміни для ділянок поверхні, базованих на параметричних кривих. Всі властивості визначені в похідних підтипах. Всі похідні підтипи мають відповідати обмеженням, визначеним у ISO 19107:2003, пункт 6.4.40.

Якщо подано атрибут `aggregationType`, то він повинен мати значення “set”.

### 10.5.12.9 AbstractGriddedSurfaceType, AbstractGriddedSurface

```
<complexType name="AbstractGriddedSurfaceType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractParametricCurveSurfaceType">
      <sequence>
        <group ref="gml:PointGrid"/>
      </sequence>
      <attribute name="rows" type="integer"/>
      <attribute name="columns" type="integer"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="AbstractGriddedSurface" type="gml:AbstractGriddedSurfaceType" abstract="true"
substitutionGroup="gml:AbstractParametricCurveSurface"/>
```

`gml:AbstractGriddedSurface` реалізує геометричний об’єкт регулярно поділена поверхня `GM_GriddedSurface`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.41). Якщо подано атрибут `rows`, то він дає число рядків, `columns` – число колонок у параметрі мережі, на які поділена поверхня. Параметр “мережа” подають через екземпляр групи `gml:PointGrid`.

Цей елемент надає голову групи заміни для ділянок поверхні, базованих на мережі. Всі похідні підтипи також мають узгоджуватися з обмеженнями, визначеними в ISO 19107:2003, пункт 6.4.41.

### 10.5.12.10 ConeType, Cone

```
<complexType name="ConeType">
  <complexContent>
    <extension base="gml:AbstractGriddedSurfaceType">
      <attribute name="horizontalCurveType" type="gml:CurveInterpolationType"
fixed="circularArc3Points"/>
      <attribute name="verticalCurveType" type="gml:CurveInterpolationType"
fixed="linear"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Cone" type="gml:ConeType"
substitutionGroup="gml:AbstractGriddedSurface"/>
```

`gml:Cone` реалізує геометричний об’єкт конус `GM_Cone`, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.42).

### 10.5.12.11 CylinderType, gml:Cylinder

```
<complexType name="CylinderType">
  <complexContent>
    <extension base="gml:AbstractGriddedSurfaceType">
      <attribute name="horizontalCurveType" type="gml:CurveInterpolationType"
fixed="circularArc3Points"/>
    </extension>
  </complexContent>
</complexType>
```

```

        fixed="circularArc3Points"/>
        <attribute name="verticalCurveType" type="gml:CurveInterpolationType"
        fixed="linear"/>
    </extension>
</complexContent>
</complexType>

```

```

<element name="Cylinder" type="gml:CylinderType"
substitutionGroup="gml:AbstractGriddedSurface"/>

```

gml:Cylinder реалізує геометричний об'єкт циліндр GM\_Cylinder, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.43).

#### 10.5.12.12 SphereType, Sphere

```

<complexType name="SphereType">
    <complexContent>
        <extension base="gml:AbstractGriddedSurfaceType">
            <attribute name="horizontalCurveType" type="gml:CurveInterpolationType"
            fixed="circularArc3Points"/>
            <attribute name="verticalCurveType" type="gml:CurveInterpolationType"
            fixed="circularArc3Points"/>
        </extension>
    </complexContent>
</complexType>

```

```

<element name="Sphere" type="gml:SphereType"
substitutionGroup="gml:AbstractGriddedSurface"/>

```

gml:Sphere реалізує геометричний об'єкт сфера GM\_Sphere, визначений в ISO 19107 (див. Г.2.3.4 та ISO 19107:2003, пункт 6.4.44).

### 10.6 Трьохвимірні геометричні примітиви

#### 10.6.1 AbstractSolidType, AbstractSolid

```

<complexType name="AbstractSolidType">
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType"/>
    </complexContent>
</complexType>

```

```

<element name="AbstractSolid" type="gml:AbstractSolidType" abstract="true"
substitutionGroup="gml:AbstractGeometricPrimitive" />

```

gml:AbstractSolidType є абстракцією тіла для підтримки різних рівнів складності. Тіло завжди можна розглядати як геометричний примітив, тобто воно є суміжним.

Елемент gml:AbstractSolid є абстрактним головою групи заміни для всіх (безперервних) елементів, що є тілами.

#### 10.6.2 SolidPropertyType, solidProperty

```

<complexType name="SolidPropertyType">
    <sequence minOccurs="0">
        <element ref="gml:AbstractSolid"/>
    </sequence>

```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Властивість, для якої тіло є її доменом значень, може або бути відповідним елементом геометрії, інкапсульованим в елементі цього типу, або бути XLink посиланням на віддалений елемент геометрії (де віддаленими є елементи геометрії, розташовані десь ще у тому ж самому документі). Треба обов'язково подавати тільки елемент, що міститься, або тільки посилання на нього.

```
<element name="solidProperty" type="gml:SolidPropertyType"/>
```

Цей елемент властивості або пов'язаний з тілом через XLink атрибути, або містить елемент, що є тілом.

gml:solidProperty є наперед визначеною властивістю, яку можна використовувати в GML прикладній схемі завжди, коли GML об'єкт має властивість із значенням, яким можна замінити AbstractSolid.

### 10.6.3 SolidArrayType, solidArrayProperty

```

<complexType name="SolidArrayType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="gml:AbstractSolid" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

gml:SolidArrayType є вмістилищем для масиву тіл. Елементи завжди містяться у властивості масиву, а посилання на елементи чи масиви геометрії не підтримуються.

```
<element name="solidArrayProperty" type="gml:SolidArrayType"/>
```

Цей елемент властивості містить список елементів, які є тілами. solidArrayProperty є наперед визначеною властивістю, яку завжди можна використовувати в GML прикладних схемах, якщо GML географічний об'єкт має властивість із значенням, яким можна замінити список gml:AbstractSolid.

### 10.6.4 SolidType, Solid

```

<complexType name="SolidType">
  <complexContent>
    <extension base="gml:AbstractSolidType">
      <sequence>
        <element name="exterior" type="gml:ShellPropertyType" minOccurs="0"/>
        <element name="interior" type="gml:ShellPropertyType" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```
<element name="Solid" type="gml:SolidType" substitutionGroup="gml:AbstractSolid" />
```

`gml:Solid` впроваджує геометричний об'єкт, визначений в ISO 19107 `GM_Solid` (див. Г.2.3.3 та ISO 19107:2003, пункт 6.3.18).

Тіло є базисом для трьохвимірної геометрії. Екстенст тіла визначений граничними поверхнями (див. ISO 19107:2003, пункт 6.3.18). `gml:exterior` визначає зовнішні межі, а `gml:interior` – внутрішні межі тіла.

### 10.6.5 ShellType, Shell

```
<complexType name="ShellType">
  <sequence>
    <element ref="gml:surfaceMember" maxOccurs="unbounded"/>
  </sequence>
  <attributeGroup ref="gml:AggregationAttributeGroup"/>
</complexType>

<element name="Shell" type="gml:ShellType" substitutionGroup="gml:AbstractObject"/>
<element name="gml:surfaceMember" type="gml:SurfacePropertyType"/>
```

`gml:Shell` реалізує геометричний об'єкт оболонка `GM_Shell`, визначений в ISO 19107 (див. Г.2.3.3 та ISO 19107:2003, пункт 6.3.8).

Оболонку використовують як єдиний зв'язний компонент межі тіла, як визначено в ISO 19107:2003, пункт 6.3.8.

Кожний `gml:surfaceMember` містить одну поверхню чи посилається на неї, тобто кожним елементом може замінити `gml:AbstractSurface`. У контексті оболонки поверхні описують межу тіла.

Якщо надано атрибут `aggregationType`, то він повинен мати значення “set”.

**Примітка 1.** Це визначення дозволяє вкладену структуру, тобто `surfaceMember` може бути композитною поверхнею `gml:CompositeSurface`, яку, в свою чергу, можна побудувати з інших композитних поверхонь `gml:CompositeSurfaces` як поверхонь, що є її членами.

### 10.6.6 ShellPropertyType

```
<complexType name="ShellPropertyType">
  <sequence>
    <element ref="gml:Shell"/>
  </sequence>
</complexType>
```

Властивість із цією моделлю контенту типу `gml:ShellPropertyType` інкапсулює оболонку для подання компоненту межі тіла.

## 11 GML СХЕМА – ГЕОМЕТРИЧНИЙ КОМПЛЕКС, ГЕОМЕТРИЧНІ КОМПЗИТИ ТА ГЕОМЕТРИЧНІ АГРЕГАТИ

### 11.1 Загальний огляд

Цей розділ описує компоненти схеми геометрії для геометричних комплексів та агрегатів.

**Примітка 1.** Документи схеми геометрії `geometryAggregates.xsd` та `geometryComplexes.xsd` (див. додаток В) ідентифіковані через незалежні від розміщення імена (з допомогою URN синтаксису):

`urn:x-ogc:specification:gml:schema-xsd:geometryAggregates:3.2.1`

urn:x-ogc:specification:gml:schema-xsd:geometryComplexes:3.2.1

Геометричні агрегати (тобто, екземпляри підтипу типу `gml:AbstractGeometricAggregateType`) є довільними агрегаціями елементів геометрії. Немає припущення, що вони мають будь-яку додаткову внутрішню структуру та використовуються для “збору” часток геометрії певного типу. У прикладних схемах можна використовувати агрегати для географічних об’єктів, для подання яких використовують множинні геометричні об’єкти.

Геометричні комплекси (тобто екземпляри типу `gml:GeometricComplexType`) є замкненими колекціями геометричних примітивів, тобто вони містять власні межі.

Геометричний комплекс (`gml:GeometricComplex`) визначений в ISO 19107:2003, пункт 6.6.1 як “набір примітивних геометричних об’єктів (в єдиній системі координат), що не є внутрішньо суцільним. Далі, якщо примітив є в геометричному комплексі, то існує набір примітивів у цьому комплексі, точкове об’єднання яких є межею цього першого примітиву”.

Геометричний композит (`gml:CompositeCurve`, `gml:CompositeSurface` та `gml:CompositeSolid`) представляє геометричний комплекс з базовою ключовою геометрією, ізоморфною по відношенню до примітиву, тобто його можна розглядати як примітив і як комплекс. Дивіться більш докладно про природу геометрій композитів у ISO 19107:2003, підрозділ. 6.1 та пункт 6.6.3.

Геометричні комплекси та композити призначені для використання у прикладних схемах, де зв’язність геометрії є важливою.

## 11.2 Геометричні комплекси та геометричні композити

### 11.2.1 Геометричний комплекс

#### 11.2.1.1 `GeometricComplexType`, `GeometricComplex`

```
<complexType name="GeometricComplexType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <element name="element" type="gml:GeometricPrimitivePropertyType"
          maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="GeometricComplex" type="gml:GeometricComplexType"
substitutionGroup="gml:AbstractGeometry" />
```

`gml:GeometricComplex` реалізує `GM_Complex`, визначений в ISO 19107 (див. ISO 19107:2003, пункти 6.6.2 та 6.6.1), згідно до Г.2.3.6.

`gml:element` містить один геометричний примітив (у тому числі з композитною геометрією) або посилається на нього (це включає композитні геометрії).

#### 11.2.1.2 `GeometricComplexPropertyType`

```
<complexType name="GeometricComplexPropertyType">
  <sequence minOccurs="0">
```



```

<choice>
  <element ref="gml:GeometricComplex"/>
  <element ref="gml:CompositeCurve"/>
  <element ref="gml:CompositeSurface"/>
  <element ref="gml:CompositeSolid"/>
</choice>
</sequence>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

Властивість, домен значень якої є геометричним комплексом, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (віддалені включають елементи геометрії, розміщені десь іще у тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

**Примітка 1.** Допустимі елементи геометрії, що містяться в такій властивості (або на які в ній спрямовані посилання), моделюють через елемент вибору XML схеми, оскільки композити (концептуально) успадковують і геометричний комплекс, і геометричний примітив, та вже є частиною групи заміни gml:AbstractGeometricPrimitive.

## 11.2.2 Геометрії композиту

### 11.2.2.1 Загальне подання композитів у GML

Члени геометричного композиту мають являти однорідну колекцію геометричних примітивів, об'єднання яких було б ключовою геометрією композиту. Комплекс має включати всі примітиви, що є членами, та всі примітиви на межі цих примітивів, доки не включені gml:Points. Тому властивості у gml:CompositeCurve, gml:CompositeSurface та gml:CompositeSolid являють піднабір властивості gml:element, яка характеризує gml:GeometricComplex.

Оскільки XML схема не підтримує концепцію “множинного успадкування”, використану в ISO 19107, для подання подвійності геометричних композитів (як відкритий примітив та як замкнений комплекс) у GML схемі композити походять тільки від gml:AbstractGeometricPrimitiveType. Однак, через використання елемента <choice> у типі властивості gml:GeometricComplexPropertyType, композит можна використати в будь-якій властивості, яка передбачає gml:GeometricComplex як своє значення.

#### 11.2.2.2 CompositeCurveType, CompositeCurve

```

<complexType name="CompositeCurveType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <element ref="gml:curveMember" maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

```
<element name="CompositeCurve" type="gml:CompositeCurveType"
substitutionGroup="gml:AbstractCurve" />
```

`gml:CompositeCurve` реалізує геометричний об'єкт композитна крива `GM_CompositeCurve`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.6.5), згідно до Г.2.3.6.

`gml:CompositeCurve` подана послідовністю кривих (які можна орієнтувати), де кожна крива в послідовності закінчується в початковій точці наступної кривої в списку.

`gml:curveMember` містить як включену одну криву в складі композитної кривої або посилається на неї.

Криві є суміжними, а колекція кривих - упорядкованою. Тому, якщо подано атрибут `aggregationType`, він матиме значення "sequence".

**Примітка 1.** Це визначення допускає вкладену структуру, тобто у композитній кривій `gml:CompositeCurve` можна використовувати, наприклад, іншу композитну криву `gml:CompositeCurve` як криву, що є членом.

### 11.2.2.3 CompositeSurfaceType, CompositeSurface

```
<complexType name="CompositeSurfaceType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:surfaceMember" maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="CompositeSurface" type="gml:CompositeSurfaceType"
substitutionGroup="gml:AbstractSurface" />
```

`gml:CompositeSurface` реалізує геометричний об'єкт композитна поверхня `GM_CompositeSurface`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.6.6), згідно до Г.2.3.6.

`gml:CompositeSurface` подають як набір поверхонь, що можуть бути орієнтовані. Це є тип геометрії з усіма геометричними властивостями (примітивної) поверхні. Суттєвим є те, що композитна поверхня є колекцією поверхонь, що з'єднані попарно через спільні граничні криві, та які формують єдину поверхню, коли їх розглядають як єдине ціле.

`gml:surfaceMember` містить одну поверхню в композитній поверхні як включену або посилається на неї.

Поверхні є суміжними.

**Примітка 1.** Це визначення допускає вкладені структури, тобто у композитній поверхні `gml:CompositeSurface` можна використовувати, наприклад, іншу композитну поверхню `gml:CompositeSurface` як поверхню, що є членом.

### 11.2.2.4 CompositeSolidType, CompositeSolid

```
<complexType name="CompositeSolidType">
  <complexContent>
```

```

<extension base="gml:AbstractSolidType">
  <sequence>
    <element ref="gml:solidMember" maxOccurs="unbounded" />
  </sequence>
  <attributeGroup ref="gml:AggregationAttributeGroup"/>
</extension>
</complexContent>
</complexType>

```

```

<element name="CompositeSolid" type="gml:CompositeSolidType"
substitutionGroup="gml:AbstractSolid" />

```

`gml:CompositeSolid` реалізує геометричний об'єкт композитне тіло `GM_CompositeSolid`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.6.7), згідно до Г.2.3.6.

`gml:CompositeSolid` подають як набір поверхонь, які можна орієнтувати. Це є геометричним типом з усіма геометричними властивостями (примітивного) тіла. Суттєво, що композитне тіло є колекцією тіл, що попарно з'єднані через спільні граничні поверхні та які формують єдине тіло, якщо їх розглядати як єдине ціле.

`gml:solidMember` містить одне тіло в композитному тілі або посилається на нього. Ці тіла є суміжними.

**Примітка 1.** Це визначення допускає вкладену структуру, тобто у композитному тілі `gml:CompositeSolid` можна використовувати, наприклад, інше композитне тіло `gml:CompositeSolid` як член.

### 11.3 Геометричні агрегати

#### 11.3.1 Агрегати невизначеної розмірності

##### 11.3.1.1 `AbstractGeometricAggregateType`, `AbstractGeometricAggregate`

```

<complexType name="AbstractGeometricAggregateType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

```

<element name="AbstractGeometricAggregate" type="gml:AbstractGeometricAggregateType"
abstract="true" substitutionGroup="gml:AbstractGeometry" />

```

`gml:AbstractGeometricAggregate` реалізує геометричний об'єкт агрегат `GM_Aggregate`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.5.2), згідно з Г.2.3.5. Він є абстрактним головою групи заміни для всіх геометричних агрегатів.

##### 11.3.1.2 `MultiGeometryType`, `MultiGeometry`, `geometryMember`, `geometryMembers`

```

<complexType name="MultiGeometryType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>

```

```

    <element ref="gml:geometryMember" minOccurs="0" maxOccurs="unbounded" />
    <element ref="gml:geometryMembers" minOccurs="0" />
  </sequence>
</extension>
</complexContent>
</complexType>

```

```

<element name="MultiGeometry" type="gml:MultiGeometryType"
substitutionGroup="gml:AbstractGeometricAggregate" />

```

`gml:MultiGeometry` є колекцією одного чи більше GML об'єктів геометрії довільного типу (див. Г.3.8).

Члени геометричного агрегату можна визначити або через “стандартну” властивість (`gml:geometryMember`), або через властивість масиву (`gml:geometryMembers`). Є також дійсним використання “стандартних” властивостей разом із властивостями масиву в одній і тій же колекції.

**Примітка 1.** Властивості масиву не можуть посилатися на віддалені елементи геометрії через XLink.

```

<element name="geometryMember" type="gml:GeometryPropertyType" />

```

Цей елемент властивості або містить елемент геометрії, або посилається на нього через XLink атрибути.

```

<element name="geometryMembers" type="gml:GeometryArrayPropertyType" />

```

Цей елемент властивості містить список елементів геометрії. Порядок елементів є значимим і має бути збережений під час обробки масиву.

### 11.3.1.3 MultiGeometryPropertyType, multiGeometryProperty

```

<complexType name="MultiGeometryPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeometricAggregate"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Властивість, доменом значень якої є геометричний агрегат, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (віддалені включають елементи геометрії, розміщені десь іще у тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання.

```

<element name="multiGeometryProperty" type="gml:MultiGeometryPropertyType" />

```

Цей елемент властивості або посилається на геометричний агрегат через XLink атрибути, або містить елемент “мультигеометрії”. `multiGeometryProperty` є наперед визначеною властивістю, яку можна використовувати в GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити `AbstractGeometricAggregate`.

## 11.3.2 Нульвимірні агрегати

### 11.3.2.1 MultiPointType, MultiPoint, pointMember, pointMembers

```
<complexType name="MultiPointType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:pointMember" minOccurs="0" maxOccurs="unbounded" />
        <element ref="gml:pointMembers" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="MultiPoint" type="gml:MultiPointType"
substitutionGroup="gml:AbstractGeometricAggregate" />
```

`gml:MultiPoint` реалізує геометричний об'єкт мультиточка `GM_MultiPoint`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.5.4), згідно з Г.2.3.5.

А `gml:MultiPoint` складається з одного чи більше `gml:Points`.

Члени геометричного агрегату можна визначити з допомогою або “стандартної” властивості (`gml:pointMember`), або властивості масиву (`gml:pointMembers`). Дійсним є також використання “стандартної” властивості разом із властивістю масиву в тій же самій колекції.

**Примітка 1.** Властивості масиву не можуть посилатися на віддалені елементи геометрії через XLink.

```
<element name="pointMember" type="gml:PointPropertyType" />
```

Цей елемент властивості або посилається на точку `Point` через XLink атрибути, або містить елемент `Point` (точка).

```
<element name="pointMembers" type="gml:PointArrayPropertyType" />
```

Цей елемент властивості містить список точок. Порядок елементів є значущим та має бути збережений під час обробки масиву.

### 11.3.2.2 MultiPointPropertyType, multiPointProperty

```
<complexType name="MultiPointPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:MultiPoint"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Властивість, доменом значень якої є колекція точок, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (віддалені включають елементи геометрії, розміщені в тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання.

```
<element name="multiPointProperty" type="gml:MultiPointPropertyType" />
```

Цей елемент властивості або посилається на точковий агрегат через XLink атрибути, або містить елемент "multipoint" (мультиточка). multiPointProperty є наперед визначеною властивістю, яку можна використовувати в GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити MultiPoint.

### 11.3.3 Одновимірні агрегати

#### 11.3.3.1 MultiCurveType, multiCurve, curveMembers

```
<complexType name="MultiCurveType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:curveMember" minOccurs="0" maxOccurs="unbounded" />
        <element ref="gml:curveMembers" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="MultiCurve" type="gml:MultiCurveType"
substitutionGroup="gml:AbstractGeometricAggregate" />
```

gml:MultiCurve реалізує геометричний об'єкт мультикрива GM\_MultiCurve, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.5.5), згідно з Г.2.3.5. gml:MultiCurve визначається однією чи більше абстрактних кривих gml:AbstractCurves.

Члени геометричного агрегату можна визначити з допомогою або "стандартної" властивості (gml:curveMember), або властивості масиву (gml:curveMembers). Дійсним є також використання "стандартної" властивості разом із властивостями масиву в тій же самій колекції.

**Примітка 1.** Властивості масиву не можуть посилатися на віддалені геометричні елементи через XLink.

```
<element name="curveMembers" type="gml:CurveArrayPropertyType" />
```

Цей елемент властивості містить список кривих. Порядок елементів є значущим та має зберігатися під час обробки масиву.

**Примітка 2.** gml:curveMember задекларовано в підпункті 10.5.11.

#### 11.3.3.2 MultiCurvePropertyType, multiCurveProperty

```
<complexType name="MultiCurvePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:MultiCurve" />
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup" />
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
```

Властивість, доменом значень якої є колекція кривих, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на

віддалений елемент геометрії (де віддалені включають елементи геометрії, розташовані десь іще в тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

```
<element name="multiCurveProperty" type="gml:MultiCurvePropertyType" />
```

Цей елемент властивості або посилається на криволінійний агрегат через XLink атрибути, або містить елемент “мультикрива”. gml:multiCurveProperty є наперед визначеною властивістю, яку можна використовувати в GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити gml:MultiCurve.

### 11.3.4 Двохвимірні агрегати

#### 11.3.4.1 MultiSurfaceType, MultiSurface, surfaceMember, surfaceMembers

```
<complexType name="MultiSurfaceType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:surfaceMember" minOccurs="0" maxOccurs="unbounded" />
        <element ref="gml:surfaceMembers" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="MultiSurface" type="gml:MultiSurfaceType"
substitutionGroup="gml:AbstractGeometricAggregate" />
```

gml:MultiSurface реалізує геометричний об'єкт мультиповерхня GM\_MultiSurface, визначений у стандарті ISO 19107 (див. ISO 19107:2003, пункт 6.5.6), згідно до Г.2.3.5. gml:MultiSurface визначено однією чи більше абстрактними поверхнями gml:AbstractSurfaces.

Члени геометричного агрегату можна визначити з допомогою або “стандартної” властивості (gml:surfaceMember), або властивості масиву (gml:surfaceMembers). Є також дійсним використання “стандартної” властивості разом із властивостями масиву в тій же самій колекції.

**Примітка 1.** Властивості масиву не можуть посилатися на віддалені елементи геометрії через XLink.

```
<element name="surfaceMember" type="gml:SurfacePropertyType" />
```

Цей елемент властивості або посилається на поверхню через атрибути XLink, або містить елемент поверхні. Елемент поверхні – це будь-який елемент, яким можна замінити абстрактну поверхню gml:AbstractSurface.

```
<element name="surfaceMembers" type="gml:SurfaceArrayPropertyType" />
```

Цей елемент властивості містить список поверхонь. Порядок елементів є значущим та має бути збережений під час обробки масиву.

#### 11.3.4.2 MultiSurfacePropertyType, multiSurfaceProperty

```
<complexType name="MultiSurfacePropertyType">
```

```

<sequence minOccurs="0">
  <element ref="gml:MultiSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Властивість, доменом значень якої є колекція поверхонь, може бути або елементом геометрії, інкапсульованим у елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалені включають елементи геометрії, розміщені десь іще в тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

```

<element name="multiSurfaceProperty" type="gml:MultiSurfacePropertyType" />

```

Цей елемент властивості або посилається на агрегат поверхонь через XLink атрибуту, або містить “мультиповерхневий” елемент. `gml:multiSurfaceProperty` і є наперед визначеною властивістю, яку можна використовувати в GML прикладних схемах завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити `gml:MultiSurface`.

### 11.3.5 Трьохвимірні агрегати

#### 11.3.5.1 MultiSolidType, MultiSolid, solidMember, solidMembers

```

<complexType name="MultiSolidType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:solidMember" minOccurs="0" maxOccurs="unbounded" />
        <element ref="gml:solidMembers" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<element name="MultiSolid" type="gml:MultiSolidType"
substitutionGroup="gml:AbstractGeometricAggregate" />

```

`gml:MultiSolid` реалізує геометричний об'єкт мультитіло `GM_MultiSolid`, визначений в ISO 19107 (див. ISO 19107:2003, пункт 6.5.7), згідно з Г.2.3.5.

`gml:MultiSolid` визначено одними чи більше абстрактними тілами `gml:AbstractSolid`.

Члени геометричного агрегату можна визначити з допомогою або “стандартної” властивості (`gml:solidMember`), або властивості масиву (`gml:solidMembers`). Є дійсним використання “стандартної” властивості разом із властивостями масиву в тій же самій колекції.

**Примітка 1.** Властивості масиву не можуть посилатися на віддалені елементи геометрії через XLink.

```

<element name="solidMember" type="gml:SolidPropertyType" />

```



Цей елемент властивості або посилається на тіло через XLink атрибути, або містить елемент, що є тілом. Цей елемент є будь-яким елементом, яким можна замінити `gml:AbstractSolid`.

```
<element name="solidMembers" type="gml:SolidArrayPropertyType" />
```

Цей елемент містить список тіл. Порядок елементів є значущим та має бути збережений під час обробки масиву.

### 11.3.5.2 MultiSolidPropertyType, multiSolidProperty

```
<complexType name="MultiSolidPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:MultiSolid"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Властивість, доменом значень якої є колекція тіл, може бути або відповідним елементом геометрії, інкапсульованим в елементі цього типу, або XLink посиланням на віддалений елемент геометрії (де віддалені включають елементи геометрії, розміщені десь ще у тому ж самому документі). Треба обов'язково подавати або тільки елемент, що міститься, або тільки посилання на нього.

```
<element name="multiSolidProperty" type="gml:MultiSolidPropertyType" />
```

Цей елемент властивості або посилається на агрегат тіл через XLink атрибути, або містить елемент “мультитіло”. `gml:multiSolidProperty` є наперед визначеною властивістю, яку можна використовувати в GML прикладній схемі завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити `gml:MultiSolid`.

## 12 GML СХЕМА – СХЕМИ РЕФЕРЕНЦІЙНИХ СИСТЕМ КООРДИНАТ

### 12.1 Огляд

#### 12.1.1 Вступ

У цьому розділі описані компоненти GML схеми для кодування визначень референційних систем координат та координатних операцій, та подано пояснення їх змісту, структури та залежностей.

#### 12.1.2 Зв'язок з ISO 19111

Компоненти схеми в GML схемі, описані в цьому розділі, надають узгоджену та повну реалізацію концептуальної схеми, визначеної в ISO 19111. Зв'язок докладно розглянутий в Г.2.7. Додаткові компоненти для часових референційних систем визначені в Г.3.9.

Типи визначені в ISO 19111, реалізовані в GML; деякі додаткові обмеження визначені в ISO 19111 для цих типів, що є обмеженнями на компоненти схеми в GML схемі.

**Примітка 1.** Відповідні п'ять документів схеми включають `referenceSystems.xsd`, `coordinateReferenceSystems.xsd`, `datums.xsd`, `coordinateSystems.xsd`, та `coordinateOperations.xsd`. Ці документи схеми реалізовані в пакеті UML із подібним ім'ям у концептуальній моделі.

### 12.1.3 Важливі XML елементи

Цими компонентами XML схеми закодовано дані для визначення референцних систем координат (РСК) та координатних операцій (включно із координатними трансформаціями та перетвореннями). Дані для визначення включають дані ідентифікації та специфікації, як це потрібно. Дивіться семантику компонентів схеми в ISO 19111.

Визначене XML кодування включає множинні альтернативні XML елементи вищого рівня, які можна використовувати, до це потрібно (тобто, немає одного-єдиного елементу вищого рівня, що може бути базисом для всіх XML документів). Більшість цих XML елементів вищого рівня є GML об'єктами, які включають ідентифікаційну інформацію, що дозволяє на них посылатись. Альтернативні XML елементи вищого рівня включають:

- всі конкретні XML елементи у групі заміни, очолюваної абстрактним XML елементом SingleCRS (ці елементи можна використовувати для передачі визначення однієї референцної системи координат цього типу), а саме: CompoundCRS (компонована референцна система координат), GeodeticCRS, ProjectedCRS, EngineeringCRS, ImageCRS (референцна система координат зображень), VerticalCRS, TemporalCRS (часова референцна система координат), DerivedCRS (похідна референцна система координат);

- всі конкретні XML елементи, якими можна замінити абстрактний XML елемент CoordinateOperation, а саме: ConcatenatedOperation, PassThroughOperation, Transformation, Conversion;

- у конкретних XML елементах, якими можна замінити SingleCRS, використовують множинні XML елементи нижчого рівня, які містять структури даних; ці елементи нижчого рівня включають усі п'ять конкретних елементів, якими можна замінити абстрактний XML елемент Datum, а саме: GeodeticDatum, VerticalDatum, TemporalDatum, EngineeringDatum, ImageDatum; ці XML елементи нижчого рівня також включають усі десять конкретних елементів, якими можна замінити абстрактний XML елемент CoordinateSystem, а саме: EllipsoidalCS, VerticalCS, CartesianCS, AffineCS, LinearCS, PolarCS, SphericalCS, CylindricalCS, TimeCS (часова координатна система), UserDefinedCS (координатна система, визначена користувачем).

- у конкретних XML елементах, якими можна замінити елемент CoordinateOperation, використовують множинні елементи нижчого рівня, що включають структури даних, включно з елементами, а саме: OperationMethod, OperationParameter, OperationParameterGroup, ParameterValue (значення параметру), ParameterValueGroup.

## 12.2 Референцні системи

### 12.2.1 Огляд

Компоненти схеми референцних систем мають дві логічні частини, які визначають елементи та типи для XML кодування визначень:

- ідентифікований об'єкт, успадкований десятьма типами GML об'єктів, використаними для референцних систем координат та координатних операцій;

- частина визначень (високий рівень) референцних систем координат.

У цій схемі закодовані пакети Ідентифікований об'єкт та Референцна Система у моделі UML для ISO 19111.

**Примітка 1.** Схема referenceSystems включає документ GML схеми dictionary.xsd, та імпортує документ схеми metadataEntitySet.xsd із стандарту ISO 19139. Цей документ схеми ідентифікований таким іменем (незалежним від розміщення, з допомогою URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:referenceSystems:3.2.1

### 12.2.2 IdentifiedObjectType

```
<complexType name="IdentifiedObjectType" abstract="true">
  <complexContent>
    <extension base="gml:DefinitionType"/>
  </complexContent>
</complexType>
```

gml:IdentifiedObjectType забезпечує властивості ідентифікації об'єкту, пов'язаного з референчною системою координат. gml:identifier має бути первинним іменем у gml:DefinitionType, за яким ідентифікують цей об'єкт, кодуючи атрибут "name" у UML моделі.

Нуль чи більше елементів gml:name можуть бути невпорядкованим набором "ідентифікаторів", кодуючи атрибут "identifier" ("ідентифікатор") у UML моделі. Кожен із цих елементів gml:name може посилатись на інформацію про визначення об'єкту або бути ідентифікатором, через який спрямоване посилання на цей об'єкт.

Нуль чи більше інших елементів gml:name можуть бути невпорядкованим набором альтернативних імен, за якими ідентифікують цей об'єкт, пов'язаний із референчною системою координат, кодуючи атрибути "alias" (альтернативне ім'я) у UML моделі. Об'єкт може мати кілька альтернативних імен, які звичайно використовують у різних контекстах. Контекст для альтернативного імені ідентифікують за значенням його (необов'язкового) атрибуту codeSpace.

Будь-яку потрібну інформацію про версію треба включати в атрибут codeSpace, що належить до gml:identifier, та елементи gml:name. У цьому випадку елемент gml:remarks у gml:DefinitionType має містити коментарі або іншу інформацію про цей об'єкт, у тому числі інформацію про джерело даних.

### 12.2.3 Абстрактна референчна система координат

#### 12.2.3.1 AbstractCRS

```
<element name="AbstractCRS" type="gml:AbstractCRSType" abstract="true"
substitutionGroup="gml:Definition"/>

<complexType name="AbstractCRSType" abstract="true">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:domainOfValidity" minOccurs="0"
maxOccurs="unbounded"/>
        <element ref="gml:scope" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:AbstractCRS` визначає референцну систему координат, що звичайно є одинарною, але може бути компонованою.

Абстрактний комплексний тип має бути використаний, розширений або обмежений, у GML прикладній схемі для визначення конкретного підтипу із смислом, еквівалентним конкретному підтипу, визначеного в цьому документі.

#### 12.2.3.2 domainOfValidity

```
<element name="domainOfValidity">
  <complexType>
    <sequence minOccurs="0">
      <element ref="gmd:EX_Extent"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</element>
```

Властивість `gml:domainOfValidity` реалізує роль в асоціації до об'єкту `EX_Extent`, як заковано в ISO/TS 19139, або включаючи визначення цього екстену, або посилаючись на нього.

#### 12.2.3.3 scope

```
<element name="scope" type="string"/>
```

Властивість `gml:scope` забезпечує опис цього використання або обмежень на використання, для яких цей об'єкт, пов'язаний із референчною системою координат (`CRSrelated`), є дійсним. Якщо значення невідоме, треба вводити "not known".

#### 12.2.3.4 CRSPropertyType

```
<complexType name="CRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:CRSPropertyType` є типом властивості для ролей в асоціації до абстрактної референчної системи координат, що або включає визначення цієї РСК, або посилається на неї.

### 12.3 Референцні системи координат

#### 12.3.1 Огляд

Компоненти схем просторово-часових референцних систем координат розподілені у дві логічні груп. Одна група визначає елементи і типи визначень абстрактних референцних систем координат для XML кодування. Більша група визначає спеціалізовані конструкції для XML кодування визначень просторово-часових референцних систем координат.

У цих компонентах схеми заковані пакети референцних систем координат у UML моделях стандарту ISO 19111 (розділ 8 та and Г.3.10 цього міжнародного стандарту, за виключенням абстрактного класу "SC\_CRS").

**Примітка 1.** Документ схеми coordinateReferenceSystems включає документи GML схеми coordinateSystems.xsd, datums.xsd та coordinateOperations.xsd. Документ схеми ідентифікують через ім'я, незалежне від розміщення (з використанням синтаксису):

urn:ogc:x-specification:gml:schema-xsd:coordinateReferenceSystems:3.2.1

## 12.3.2 Абстрактні референційні системи координат

### 12.3.2.1 AbstractSingleCRS

```
<element name="AbstractSingleCRS" type="gml:AbstractCRSType" abstract="true"
substitutionGroup="gml:AbstractCRS"/>
```

gml:AbstractSingleCRS реалізує референційну систему координат, що містить одну систему координат та одну дату (на відміну від компонованої референційної системи координат).

### 12.3.2.2 SingleCRSPropertyType

```
<complexType name="SingleCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractSingleCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:SingleCRSPropertyType є типом властивості для ролей в асоціації до одинарної системи координат, що або містить визначення референційної системи координат, або посилається на нього.

### 12.3.2.3 AbstractGeneralDerivedCRS

```
<element name="AbstractGeneralDerivedCRS" type="gml:AbstractGeneralDerivedCRSType"
abstract="true" substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="AbstractGeneralDerivedCRSType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <element ref="gml:conversion"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

gml:AbstractGeneralDerivedCRS є референційною системою координат, що визначена своїм перетворенням координат з іншої референційної системи координат. Абстрактний комплексний тип не можна використовувати, розширювати або обмежувати у GML прикладній схемі задля визначення конкретного підтипу із смислом, еквівалентним конкретному підтипу, визначеному в цьому документі.

### 12.3.2.4 conversion

```
<element name="conversion" type="gml:GeneralConversionPropertyType"/>
```

gml:conversion є роллю в асоціації до перетворення координат, яку використовують для визначення референційної системи координат.

### 12.3.3 Конкретні референці системи координат

#### 12.3.3.1 CompoundCRS

```
<element name="CompoundCRS" type="gml:CompoundCRSType"
substitutionGroup="gml:AbstractCRS"/>
```

```
<complexType name="CompoundCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <element ref="gml:componentReferenceSystem" minOccurs="2"
maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

gml:CompoundCRS є референчною системою координат, що описує позицію точок через дві чи більше незалежних референціальних систем координат. Вона асоційована з неповторюваною послідовністю двох чи більше екземплярів SingleCRS.

#### 12.3.3.2 ComponentReferenceSystem

```
<element name="componentReferenceSystem" type="gml:SingleCRSPropertyType"/>
```

Елементи gml:componentReferenceSystem є впорядкованою послідовністю асоціацій із усіма референціальними системами координат, що є компонентами в компонованій референціальній системі координат.

Треба використовувати gml:AggregationAttributeGroup для визначення того, що властивості, що характеризують gml:componentReferenceSystem, упорядковані.

#### 12.3.3.3 CompoundCRSPropertyType

```
<complexType name="CompoundCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:CompoundCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:CompoundCRSPropertyType є типом властивості для ролей в асоціації до компонованої референчної системи координат, що або містить визначення цієї референчної системи, або посилається на нього.

#### 12.3.3.4 GeodeticCRS

```
<element name="GeodeticCRS" type="gml:GeodeticCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="GeodeticCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
```

```

    <choice>
      <element ref="gml:ellipsoidalCS"/>
      <element ref="gml:cartesianCS"/>
      <element ref="gml:sphericalCS"/>
    </choice>
    <element ref="gml:geodeticDatum"/>
  </sequence>
</extension>
</complexContent>
</complexType>

```

gml:GeodeticCRS є референчною системою координат, базованою на геодезичній даті.

#### 12.3.3.5 ellipsoidalCS

```
<element name="ellipsoidalCS" type="gml:EllipsoidalCSPropertyType"/>
```

gml:ellipsoidalCS є роллю в асоціації до еліпсоїдної системи координат, що використовують у цій системі координат.

#### 12.3.3.6 cartesianCS

```
<element name="cartesianCS" type="gml:CartesianCSPropertyType"/>
```

gml:cartesianCS є роллю в асоціації до декартової системи координат, що використовують у цій системі координат.

#### 12.3.3.7 sphericalCS

```
<element name="sphericalCS" type="gml:SphericalCSPropertyType"/>
```

gml:sphericalCS є роллю в асоціації до сферичної системи координат, що використовують у цій системі координат.

#### 12.3.3.8 geodeticDatum

```
<element name="geodeticDatum" type="gml:GeodeticDatumPropertyType"/>
```

gml:geodeticDatum є роллю в асоціації до геодезичної дати, яку використовують в цій референційній системі координат.

*Елемент gml:usesGeodeticDatum скасований та замінений на gml:geodeticDatum.*

#### 12.3.3.9 GeodeticCRSPropertyType

```

<complexType name="GeodeticCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GeodeticCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

gml:GeodeticCRSPropertyType є типом властивості ролей в асоціації до геодезичної референційної системи координат, що містить або визначення цієї референційної системи, або посилається на нього.

#### 12.3.3.10 VerticalCRS

```
<element name="VerticalCRS" type="gml:VerticalCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```

<complexType name="VerticalCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <element ref="gml:verticalCS"/>
        <element ref="gml:verticalDatum"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:VerticalCRS` є одновимірною референчною системою координат, яку використовують для запису висот та глибин. Вона базована на напрямку гравітації для визначення концепції висоти або глибини, але відношення до гравітації може не бути прямолінійним. Еліпсоїдні висоти ( $h$ ) не можуть бути отримані в вертикальній референційній системі координат. Еліпсоїдні висоти не можуть існувати незалежно, натомість вони є невідривною частиною кортежу трьохвимірних координат, визначених у географічній трьохвимірній референційній системі координат.

#### 12.3.3.11 verticalCS

```
<element name="verticalCS" type="gml:VerticalCSPropertyType"/>
```

`gml:verticalCS` є роллю в асоціації до вертикальної системи координат, що використовують у цій референційній системі координат.

#### 12.3.3.12 verticalDatum

```
<element name="verticalDatum" type="gml:VerticalDatumPropertyType"/>
```

`gml:verticalDatum` є роллю в асоціації до вертикальної дати, що використовують у цій референційній системі координат.

#### 12.3.3.13 VerticalCRSPropertyType

```

<complexType name="VerticalCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:VerticalCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:VerticalCRSPropertyType` є типом властивості для ролей в асоціації до вертикальної референційної системи координат, що або містить визначення цієї референційної системи, або посилається на нього.

#### 12.3.3.14 ProjectedCRS

```
<element name="ProjectedCRS" type="gml:ProjectedCRSType"
substitutionGroup="gml:AbstractGeneralDerivedCRS"/>
```

```

<complexType name="ProjectedCRSType">
  <complexContent>
    <extension base="gml:AbstractGeneralDerivedCRSType">
      <sequence>

```



```

    <choice>
      <element ref="gml:baseGeodeticCRS"/>
      <element ref="gml:baseGeographicCRS"/>
    </choice>
    <element ref="gml:cartesianCS"/>
  </sequence>
</extension>
</complexContent>
</complexType>

```

`gml:ProjectedCRS` є двохвимірною референчною системою координат, яку використовують для апроксимації форми Землі на плоскій поверхні, але так, щоб викривлення, спричинені апроксимацією, були добре контрольовані та відомі. Корекцію викривлення звичайно застосовують до обчислених кутів та віддалей для отримання значень, що чітко узгоджені з дійсними польовими даними.

#### 12.3.3.15 baseGeodeticCRS

```
<element name="baseGeodeticCRS" type="gml:GeodeticCRSPropertyType"/>
```

`gml:baseGeodeticCRS` є роллю в асоціації до геодезичної референчної системи координат, використаної в цій проекційній референційній системі координат.

#### 12.3.3.16 ProjectedCRSPropertyType

```

<complexType name="ProjectedCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:ProjectedCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:ProjectedCRSPropertyType` є типом властивості для ролей в асоціації до проекційованої референчної системи координат, що або містить визначення цієї референційної системи, або посилається на нього.

#### 12.3.3.17 DerivedCRS

```
<element name="DerivedCRS" type="gml:DerivedCRSType"
substitutionGroup="gml:AbstractGeneralDerivedCRS"/>
```

```

<complexType name="DerivedCRSType">
  <complexContent>
    <extension base="gml:AbstractGeneralDerivedCRSType">
      <sequence>
        <element ref="gml:baseCRS"/>
        <element ref="gml:derivedCRSType"/>
        <element ref="gml:coordinateSystem"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:DerivedCRS` є одинарною референчною системою координат, що визначена через перетворення координат з іншої одиничної референчної системи координат, відомої як базова референчна система координат. Ця базова система може бути проєкційованою референчною системою координат, якщо `DerivedCRS` використовують до GRID-покриття, прив'язаного до земної системи координат (визначене в ISO 19123, розділ 8).

#### 12.3.3.18 baseCRS

```
<element name="baseCRS" type="gml:SingleCRSPropertyType"/>
```

`gml:baseCRS` є роллю в асоціації до референчної системи координат, використаною у цій похідній референційній системі координат.

#### 12.3.3.19 derivedCRSType

```
<element name="derivedCRSType" type="gml:CodeWithAuthorityType"/>
```

Властивість `gml:derivedCRSType` описує тип похідної референційної системи координат. Необхідний атрибут `codeSpace` має забезпечувати посилання на джерело інформації, що визначає значення та смисли всіх дозволених текстових значень для цієї властивості.

#### 12.3.3.20 coordinateSystem

```
<element name="coordinateSystem" type="gml:CoordinateSystemPropertyType"/>
```

`gml:usesCS` є роллю в асоціації до системи координат, використаної у цій референційній системі координат.

*Елемент `gml:usesCS` скасований та замінений на `gml:coordinateSystem`.*

#### 12.3.3.21 DerivedCRSPropertyType

```
<complexType name="DerivedCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:DerivedCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:DerivedCRSPropertyType` є типом властивості для ролей в асоціації до непроеційованої похідної референційної системи координат, що або містить визначення цієї референційної системи координат, або посилається на нього.

#### 12.3.3.22 EngineeringCRS

```
<element name="EngineeringCRS" type="gml:EngineeringCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="EngineeringCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <choice>
          <element ref="gml:affineCS"/>
          <element ref="gml:cartesianCS"/>
          <element ref="gml:cylindricalCS"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        <element ref="gml:linearCS"/>
        <element ref="gml:polarCS"/>
        <element ref="gml:sphericalCS"/>
        <element ref="gml:userDefinedCS"/>
        <element ref="gml:coordinateSystem"/>
    </choice>
    <element ref="gml:engineeringDatum"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

`gml:EngineeringCRS` є контекстуально локальною референчною системою координат, що може належати до однієї з двох широких категорій:

- фіксовані до Землі системи, що застосовують для інженерної діяльності на поверхні Землі або біля неї;
- системи на рухомих базах, таких як засоби наземного, водного та повітряного транспорту, а також на космічних кораблях (див. ISO 19111, підрозділ 8.3).

#### **12.3.3.23 CylindricalCS**

```
<element name="CylindricalCS" type="gml:CylindricalCSPropertyType">
```

`gml:CylindricalCS` є роллю в асоціації до циліндричної системи координат, використаної в цій РСК.

#### **12.3.3.24 LinearCS**

```
<element name="LinearCS" type="gml:LinearPropertyCSType">
```

`gml:LinearCS` є роллю в асоціації до лінійної системи координат, використаної в цій РСК.

#### **12.3.3.25 PolarCS**

```
<element name="PolarCS" type="gml:PolarCSPropertyType">
```

`gml:PolarCS` є роллю в асоціації до полярної системи координат, використаної в цій РСК.

#### **12.3.3.26 UserDefinedCS**

```
<element name="UserDefinedCS" type="gml:UserDefinedCSPropertyType">
```

`gml:UserDefinedCS` є роллю в асоціації до системи координат, визначеної користувачем та використаної в цій РСК.

#### **12.3.3.27 EngineeringDatum**

```
<element name="EngineeringDatum" type="gml:EngineeringDatumPropertyType">
```

`gml:EngineeringDatum` є роллю в асоціації до інженерної дати, визначеної користувачем та використаної в цій РСК.

#### **12.3.3.28 EngineeringCRSPropertyType**

```

<complexType name="EngineeringCRSPropertyType">
    <sequence minOccurs="0">
        <element ref="gml:EngineeringCRS"/>
    </sequence>

```

```
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:EngineeringCRSPropertyType` є типом властивості для ролей в асоціації до інженерної референцної системи координат, що або містить визначення цієї референцної системи, або посилається на нього.

### 12.3.3.29 ImageCRS

```
<element name="ImageCRS" type="gml:ImageCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
<complexType name="ImageCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <choice>
          <element ref="gml:cartesianCS"/>
          <element ref="gml:affineCS"/>
          <element ref="gml:usesObliqueCartesianCS"/>
        </choice>
        <element ref="gml:imageDatum"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:ImageCRS` є інженерною референчною системою координат, що застосовують для визначення місцеположень на зображеннях. Референцні системи координат зображень розглядають як окремий підтип, оскільки визначення асоційованої дати растру містить два атрибути, недоречні для інших інженерних дат.

### 12.3.3.30 affineCS

```
<element name="affineCS" type="gml:AffineCSPropertyType"/>
```

`gml:affineCS` є роллю в асоціації до афінної референцної системи координат, що використовують у цій РСК.

### 12.3.3.31 imageDatum

```
<element name="imageDatum" type="gml:ImageDatumPropertyType"/>
```

`gml:imageDatum` є роллю в асоціації до дати зображення, що використовують у цій РСК.

### 12.3.3.32 ImageCRSPropertyType

```
<complexType name="ImageCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:ImageCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:ImageCRSPropertyType` є типом властивості для ролей в асоціації референцної системи координат зображення, що або містить визначення цієї референцної системи, або посилається на нього.

### 12.3.3.33 TemporalCRS

```
<element name="TemporalCRS" type="gml:TemporalCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="TemporalCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <choice>
          <element ref="gml:timeCS"/>
          <element ref="gml:usesTemporalCS"/>
        </choice>
        <element ref="gml:temporalDatum"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:TemporalCRS` є одновимірною референцною системою координат, що використовують для фіксації часу.

### 12.3.3.34 timeCS

```
<element name="timeCS" type="gml:TimeCSPropertyType"/>
```

`gml:timeCS` є роллю в асоціації до часової системи координат, що використовують у цій РСК.

### 12.3.3.35 temporalDatum

```
<element name="temporalDatum" type="gml:TemporalDatumPropertyType"/>
```

`gml:temporalDatum` є роллю в асоціації часової дати, що використовують у цій РСК.

### 12.3.3.36 TemporalCRSPropertyType

```
<complexType name="TemporalCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TemporalCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:TemporalCRSPropertyType` є типом властивості для ролей в асоціації часової референцної системи координат, що або містить визначення цієї референцної системи, або посилається на нього.

## 12.4 Системи координат

### 12.4.1 Огляд

Компоненти схеми систем координат розподілені в три логічні групи, які визначають елементи та типи для XML кодування визначень

- осей систем координат;
- абстрактних систем координат;
- та множинних конкретних типів просторово-часових систем координат.

Цими компонентами схеми кодують пакети систем координат UML моделей стандарту ISO 19111, розділ 9, та Г.3.10 цього міжнародного стандарту.

**Примітка 1.** Документ схеми coordinateSystems включає документ GML схеми referenceSystems.xsd. Ця схема ідентифікована таким незалежним від розміщення іменем (з використанням URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:coordinateSystems:3.2.1

## 12.4.2 Вісі системи координат

### 12.4.2.1 CoordinateSystemAxis

```
<element name="CoordinateSystemAxis" type="gml:CoordinateSystemAxisType"
  substitutionGroup="gml:Definition"/>
```

```
<complexType name="CoordinateSystemAxisType">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:axisAbbrev"/>
        <element ref="gml:axisDirection"/>
        <element ref="gml:minimumValue" minOccurs="0"/>
        <element ref="gml:maximumValue" minOccurs="0"/>
        <element ref="gml:rangeMeaning" minOccurs="0"/>
      </sequence>
      <attribute ref="uom" type="gml:UomOMIdentifier" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

gml:CoordinateSystemAxis є визначенням осі системи координат.

### 12.4.2.2 axisAbbrev

```
<element name="axisAbbrev" type="gml:CodeType"/>
```

gml:axisAbbrev є скороченням, яке використовують для цієї осі системи координат; це скорочення також використовують для ідентифікації координат у кортежі координат. Атрибут codeSpace може посилатися на джерело додаткової інформації про набір стандартних скорочень або про це скорочення.

*Приклад 1.* Типовими скороченнями є “X” та “Y”.

### 12.4.2.3 axisDirection

```
<element name="axisDirection" type="gml:CodeWithAuthorityType"/>
```

gml:axisDirection є напрямком осі цієї системи координат (або, у випадку декартових проєційованих координат, напрямком цієї осі системи координат у початку системи).

*Приклад 1.* Типовими напрямками можуть бути "north" або "south", "east" або "west", "up" або "down" (“північ”, “південь”, “схід”, “захід”, “вниз”, “вверх”).

У будь-якому наборі осей системи координат тільки одна з кожної пари може бути використана. Так, для референціальних систем координат із фіксованим напрямком на схід, цей напрямок часто є приблизним та призначений для забезпечення зрозумілого смислу цієї осі. Коли використовують геодезичну дату, то точні напрямки осей можуть через це коливатись біля цього приблизного напрямку.

**Примітка 1.** gml:EngineeringCRS часто потребує специфічних описів напрямків своїх координатних осей.

Атрибут codeSpace має посилатись на джерело інформації, яке визначає значення та смисл усіх текстових значень цієї властивості.

#### 12.4.2.4 minimumValue, maximumValue, rangeMeaning, uom

```
<element name="minimumValue" type="double"/>
<element name="maximumValue" type="double"/>
```

Властивості gml:minimumValue та gml:maximumValue дозволяють визначення мінімального та максимального значень, звичайно допустимих для цієї осі в одиницях вимірювання осі. Для безперервної кутової осі, наприклад, меридіану, значення коливаються навколо цього значення. Також значення поза мінімумом чи максимумом мають бути визначені для певної мети, наприклад, для визначення обмежувального прямокутника. Значення мінус безкінечність має бути дозволене для елемента gml:minimumValue, плюс безкінечність – для елемента gml:maximumValue. Якщо ці елементи пропущені, то значення не визначене.

```
<element name="rangeMeaning" type="gml:CodeWithAuthorityType"/>
```

gml:rangeMeaning описує смисл діапазону значень осі, визначений через gml:minimumValue та gml:maximumValue. Цей елемент треба пропустити, якщо пропущено обидва gml:minimumValue та gml:maximumValue. Цей елемент треба включати, якщо gml:minimumValue або gml:maximumValue включене. Якщо цей елемент пропущений, коли gml:minimumValue або gml:maximumValue включено, то смисл невизначений. Атрибут codeSpace має посилатися на джерело інформації, що визначає значення та смисл усіх допустимих текстових значень для цієї властивості.

#### 12.4.2.5 uom

Атрибут uom забезпечує ідентифікатор одиниці вимірювання, використаної для цієї осі системи координат. Це значення координати в кортежі координат треба записувати з використанням цієї одиниці виміру завжди, коли у цих координатах використана референціальна система координат, що використовує координатну систему з цією віссю.

#### 12.4.2.6 CoordinateSystemAxisPropertyType

```
<complexType name="CoordinateSystemAxisPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:CoordinateSystemAxis"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:CoordinateSystemAxisPropertyType є типом властивості для ролей в асоціації до осі системи координат, що або містить визначення цієї осі, або посилається на нього.

### 12.4.3 Абстрактна система координат

#### 12.4.3.1 AbstractCoordinateSystem

```
<element name="AbstractCoordinateSystem" type="gml:AbstractCoordinateSystemType"
abstract="true"
substitutionGroup="gml:Definition"/>
```

```
<complexType name="AbstractCoordinateSystemType" abstract="true">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:axis" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

`gml:AbstractCoordinateSystem` є системою координат, що є послідовністю осей системи координат (у якій немає повторів), яка охоплює даний координатний простір. Систему координат отримують з набору математичних правил для визначення способу призначення координат точкам у даному просторі. Значення координат у кортежі координат треба записувати у тому порядку, в якому записані асоціації осей системи координат.

Цей абстрактний комплексний тип можна використовувати, розширювати або обмежувати в прикладній схемі для визначення конкретного підтипу із смислом, еквівалентним конкретному підтипу, визначеному в цьому документі.

#### 12.4.3.2 axis

```
<element name="axis" type="gml:CoordinateSystemAxisPropertyType"/>
```

Властивість `gml:axis` є роллю в асоціації (упорядкована послідовність) до осей системи координат, включених у цю систему координат. Значення координат у кортежі координат треба записувати у тому порядку, в якому записані асоціації осей системи координат завжди, коли в тих координатах використано референцну систему координат, в якій застосована ця система координат. Треба використовувати `gml:AggregationAttributeGroup` для того, щоб визначити, що ці об'єкти осей упорядковані.

#### 12.4.3.3 CoordinateSystemPropertyType

```
<complexType name="CoordinateSystemPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractCoordinateSystem"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:CoordinateSystemPropertyType` є типом властивості для ролей в асоціації до системи координат, що або містить визначення цієї системи координат, або посиляється на нього.



## 2.4.4 Конкретні координатні системи

### 12.4.4.1 EllipsoidalCS

```
<element name="EllipsoidalCS"
type="gml:EllipsoidalCSType"substitutionGroup="gml:AbstractCoordinateSystem"/>

<complexType name="EllipsoidalCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

gml:EllipsoidalCS є двох- або трьохвимірною системою координат, позиція в якій визначені геодезичною широтою, геодезичною довготою та (у трьохвимірному випадку) еліпсоїдною висотою. EllipsoidalCS повинна мати два або три елементи властивості gml:axis; число асоціацій має дорівнювати розмірності системи координат.

### 12.4.4.2 EllipsoidalCSPropertyType

```
<complexType name="EllipsoidalCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:EllipsoidalCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:EllipsoidalCSPropertyType є типом властивості для ролей в асоціації до еліпсоїдної системи координат, що або містить визначення цієї системи координат, або посилання на нього.

### 12.4.4.3 CartesianCS

```
<element name="CartesianCS" type="gml:CartesianCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>

<complexType name="CartesianCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

gml:CartesianCS є одно-, двох- чи трьохвимірною системою координат. В одновимірному випадку вона містить одну пряму координатну вісь. У двох- та трьохвимірних випадках вона надає позиції точкам відносно ортогональних прямих осей. У багатовимірному випадку всі осі повинні мати одну і ту ж одиницю вимірювання. CartesianCS повинна мати один, два чи три елементів властивості gml:axis.

### 12.4.4.4 CartesianCSPropertyType

```
<complexType name="CartesianCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:CartesianCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

```
</complexType>
```

`gml:CartesianCSPropertyType` є типом властивості для ролей в асоціації до декартової системи координат, що або містить визначення цієї системи координат, або посилається на нього.

#### 12.4.4.5 VerticalCS

```
<element name="VerticalCS" type="gml:VerticalCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
```

```
<complexType name="VerticalCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

`gml:VerticalCS` є одновимірною системою координат, яку використовують для запису висот або глибин точок. Така система координат звичайно залежить від гравітаційного поля Землі, можливо, *ширше*, коли атмосферний тиск є базисом для вертикальної осі системи координат. `VerticalCS` повинна мати один елемент властивості `gml:axis`.

#### 13.4.4.6 VerticalCSPropertyType

```
<complexType name="VerticalCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:VerticalCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:VerticalCSPropertyType` є типом властивості для ролей в асоціації до вертикальної системи координат, що або містить визначення цієї системи, або посилається на нього.

#### 12.4.4.7 TimeCS

```
<element name="TimeCS" type="gml:TimeCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
```

```
<complexType name="TimeCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

`gml:TimeCS` є одновимірною системою координат, що містить часову вісь, яку використовують для позиції точки у визначених одиницях від визначеного початку в часі. `TimeCS` повинна мати один елемент властивості `gml:axis`.

#### 12.4.4.8 TimeCSPropertyType

```
<complexType name="TimeCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

```

    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:TimeCSPropertyType` є типом властивості для ролей в асоціації до часової системи координат, що або містить визначення цієї координатної системи, або посилається на нього.

#### 12.4.4.9 LinearCS

```

<element name="LinearCS" type="gml:LinearCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>

<complexType name="LinearCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>

```

`gml:LinearCS` є одновимірною системою координат, що складається з точок, які лежать на одній описаній осі. Асоційованою координатою є віддаль – із зміщенням чи без нього – від визначеної дати до точки вздовж осі. `LinearCS` повинна мати один елемент властивості `gml:axis`.

*Приклад 1.* Використання лінійного об'єкту, яким подають трубопровід для опису точок на трубопроводі чи вздовж його.

**Примітка 1.** `gml:LinearCS` можна використовувати тільки для простих безперервних лінійних систем. Лінійні референсні системи, зокрема як вони застосовані в транспортній індустрії, визначені в ISO 19133 та не реалізовані в цьому стандарті.

#### 12.4.4.10 LinearCSPropertyType

```

<complexType name="LinearCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:LinearCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:LinearCSPropertyType` є типом властивості для ролей в асоціації до лінійної системи координат, що або містить визначення цієї системи координат, або посилається на нього.

#### 12.4.4.11 UserDefinedCS

```

<element name="UserDefinedCS" type="gml:UserDefinedCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>

<complexType name="UserDefinedCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>

```

`gml:UserDefinedCS` є двох- чи трьохвимірною системою координат, що складається з будь-якої комбінації координатних осей, не охопленою будь-яким іншим типом системи координат. `UserDefinedCS` має мати два чи три елементи властивості `gml:axis`; число елементів властивості має дорівнювати розмірності системи координат.

*Приклад 1.* Мультилінійна система координат, що містить одну координатну вісь, яка може мати будь-яку одновимірну форму, не має самоперетинів. Ця непрямолінійна вісь доповнена однієї чи двома прямими осями для завершення двох- чи трьохвимірної системи координат. Непряма вісь звичайно має зростаючу кривизну чи прямолінійність.

#### 12.4.4.12 UserDefinedCSPropertyType

```
<complexType name="UserDefinedCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:UserDefinedCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:UserDefinedCSPropertyType` є типом властивості для ролей в асоціації до системи координат, визначеною користувачем, що або містить визначення цієї системи координат, або посилається на нього.

#### 12.4.4.13 SphericalCS

```
<element name="SphericalCS" type="gml:SphericalCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
```

```
<complexType name="SphericalCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

`gml:SphericalCS` є трьохвимірною системою координат із однією віддаллю, вимірною від початку системи, та двома кутовими координатами.

**Примітка 1.** Не слід плутати її з еліпсоїдною системою координат, базованою на еліпсоїді, що “дегенерував” у сферу.

`SphericalCS` повинна мати три елементи властивості `gml:axis`.

#### 12.4.4.14 SphericalCSPropertyType

```
<complexType name="SphericalCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:SphericalCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:SphericalCSPropertyType` є типом властивості для ролей в асоціації до сферичної системи координат, що або мстить визначення цієї системи координат, або посилається на нього.

**12.4.4.15 PolarCS**

```
<element name="PolarCS" type="gml:PolarCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
```

```
<complexType name="PolarCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

gml:PolarCS є двохвимірною системою координат, в якій позиція визначена віддаллю від початку системи та кутом між лінією від початку системи до точки та референцим напрямком. PolarCS повинна мати два елементи властивості gml:axis.

**12.4.4.16 PolarCSPropertyType**

```
<complexType name="PolarCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:PolarCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:PolarCSPropertyType є типом властивості для ролей в асоціації до полярної системи координат, що або містить визначення цієї системи координат, або посилається на нього.

**12.4.4.17 CylindricalCS**

```
<element name="CylindricalCS" type="gml:CylindricalCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
```

```
<complexType name="CylindricalCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

gml:CylindricalCS є трьохвимірною системою координат, що складається з полярної системи координат, доповненою прямою координатною віссю, перпендикулярною площині, охопленій полярною системою координат. CylindricalCS повинна мати три елементи властивості gml:axis.

**12.4.4.18 CylindricalCSPropertyType**

```
<complexType name="CylindricalCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:CylindricalCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:CylindricalCSPropertyType` є типом властивості для ролей в асоціації циліндричної системи координат, що або містить визначення цієї системи координат, або посилається на нього.

#### 12.4.4.19 AffineCS

```
<element name="AffineCS" type="gml:AffineCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>

<complexType name="AffineCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>
```

`gml:AffineCS` є двох- або трьохвимірною системою координат із прямими осями, які не обов'язково ортогональні. `AffineCS` повинна мати два або три елементи властивості `gml:axis`; число елементів властивості дорівнює розмірності системи координат.

#### 12.4.4.20 AffineCSPropertyType

```
<complexType name="AffineCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AffineCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:AffineCSPropertyType` є типом властивості для ролей в асоціації афінної системи координат, що або містить визначення цієї системи координат, або посилається на нього.

### 12.5 Дати

#### 12.5.1 Огляд

Компоненти схеми дат розподіляють у три логічні групи, що визначають елементи та типи для XML кодування визначень:

- абстрактних дат;
- геодезичних дат (включно із еліпсоїдом та початковим меридіаном);
- багатьох інших конкретних типів просторових та часових дат.

Цими компонентами схеми кодують пакети дат у UML моделях (ISO 19111, розділ 10 та Г.3.10 цього міжнародного стандарту).

**Примітка 1.** Документ схеми дат включає GML документ схеми `referenceSystems.xsd`. Ця схема ідентифікована таким незалежним від розміщення іменем (з допомогою URN синтаксису).

```
urn:x-ogc:specification:gml:schema-xsd:datums:3.2.1
```

#### 12.5.2 Абстрактна дата

##### 12.5.2.1 AbstractDatum

```
<element name="AbstractDatum" type="gml:AbstractDatumType" abstract="true"
substitutionGroup="gml:Definition"/>
```

```

<complexType name="AbstractDatumType" abstract="true">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:domainOfValidity" minOccurs="0"/>
        <element ref="gml:scope" maxOccurs="unbounded"/>
        <element ref="gml:anchorDefinition" minOccurs="0"/>
        <element ref="gml:realizationEpoch" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:AbstractDatum` визначає відношення системи координат до Землі, таким чином утворюючи референцну систему координат. У даті використовують набір параметрів, які визначають розташування початку референцної системи координат. Кожний підтип дати можна асоціювати тільки із специфічними типами системи координат. Цей абстрактний комплексний тип можна використовувати, розширяти та обмежувати у GML прикладній схемі для визначення конкретного підтипу з смислом, еквівалентним конкретному підтипу, визначеному в цьому документі.

#### 12.5.2.2 anchorDefinition

```
<element name="anchorDefinition" type="gml:CodeType"/>
```

`gml:anchorDefinition` є описом, можливо із включенням координат, визначення, використаного для прив'язки дати до Землі. Це також відоме як початок координат, особливо в інженерних та растрових датах. Атрибут `codeSpace` можна використовувати для посилання на джерело більш детальної інформації про цю точку про поверхню чи про набір таких описів для:

- геодезичної дати, для якої ця точка відома також як фундаментальна точка, що традиційно є точкою, в якій визначене відношення між геоїдом та еліпсоїдом; іноді “фундаментальна точка” може складатися з кількох точок (у таких випадках параметри, що визначають відношення між геоїдом та еліпсоїдом можуть бути усереднені для цих точок, а середні значення приймають як визначення дати);
- інженерної дати, для якої вихідне визначення може бути фізичною точкою, або воно може бути точкою із визначеними координатами в іншій референцній системі координат;
- для растрової дати, для якої вихідним визначенням звичайно є центр растру або кут растру;
- для часової дати цей атрибут не визначений; замість вихідного визначення, часова дата має окремий початок часу в типі `DateTime`.

#### 12.5.2.3 realizationEpoch

```
<element name="realizationEpoch" type="date"/>
```

`gml:realizationEpoch` є часом, після якого це визначення дати є дійсним (докладніше див. ISO 19111, табл. 33).

#### 12.5.2.4 DatumPropertyType

```

<complexType name="DatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:DatumPropertyType` є типом властивості для ролей в асоціації до дати, що або містить визначення цієї дати, або посилається на нього.

### 12.5.3 Геодезична дата

#### 12.5.3.1 GeodeticDatum

```

<element name="GeodeticDatum" type="gml:GeodeticDatumType"
substitutionGroup="gml:AbstractDatum"/>

```

```

<complexType name="GeodeticDatumType">
  <complexContent>
    <extension base="gml:AbstractDatumType">
      <sequence>
        <element ref="gml:primeMeridian"/>
        <element ref="gml:ellipsoid"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:GeodeticDatum` є геодезичною датою, що визначає точне розташування та орієнтацію у трьохвимірному просторі визначеного еліпсоїду (або сфери), або декартової системи координат із центром у цьому еліпсоїді (або цієї сфери).

#### 12.5.3.2 primeMeridian

```

<element name="primeMeridian" type="gml:PrimeMeridianPropertyType"/>

```

`gml:primeMeridian` є роллю в асоціації до початкового меридіану, що використовують у цій геодезичній даті.

#### 12.5.3.3 Ellipsoid

```

<element name="ellipsoid" type="gml:EllipsoidPropertyType"/>

```

`gml:ellipsoid` є роллю в асоціації еліпсоїда, що використовують у цій даті.

#### 12.5.3.4 GeodeticDatumPropertyType

```

<complexType name="GeodeticDatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GeodeticDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:GeodeticDatumPropertyType` є типом властивості для ролей в асоціації до геодезичної дати, що або містить визначення цієї дати, або посилається на нього.



### 12.5.3.5 Ellipsoid, semiMajorAxis, secondDefiningParameter

```
<element name="Ellipsoid" type="gml:EllipsoidType" substitutionGroup="gml:Definition"/>
```

```
<complexType name="EllipsoidType">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:semiMajorAxis"/>
        <element ref="gml:secondDefiningParameter"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:Ellipsoid` є геометричною фігурою, яку можна використовувати для опису апроксимованої форми Землі. В математичних термінах це є поверхня, сформована обертанням еліпсу навколо його малої осі.

```
<element name="semiMajorAxis" type="gml:MeasureType"/>
```

`gml:semiMajorAxis` визначає довжину великої напівосі еліпсоїду з її одиницями та використовує `MeasureType` з обмеженням, що одиниця вимірювання, на яку спрямоване посилання від  `uom`, має бути придатною для подання довжини, наприклад, метр або фут.

```
<element name="secondDefiningParameter">
  <complexType>
    <sequence>
      <element ref="gml:SecondDefiningParameter"/>
    </sequence>
  </complexType>
</element>
<element name="SecondDefiningParameter">
  <complexType>
    <choice>
      <element name="inverseFlattening" type="gml:MeasureType"/>
      <element name="semiMinorAxis" type="gml:LengthType"/>
      <element name="isSphere" type="boolean" default="true"/>
    </choice>
  </complexType>
</element>
```

`gml:secondDefiningParameter` є властивістю, що містить визначення другого параметру, що визначає форму еліпсоїду. Еліпсоїд потребує два визначальних параметри: велика напіввісь та зворотне вирівнювання або велика та мала напівосі. Коли референчним тілом є радше сфера, ніж еліпсоїд, то достатньо й одного визначального параметру, а саме радіусу сфери; у такому випадку велика напіввісь “дегенерує” у радіус сфери.

Елемент `inverseFlattening` містить значення зворотного сплющування еліпсоїду. Це значення є коефіцієнтом масштабу або відношенням. У ньому використано `gml:LengthType` з обмеженням, що одиниця вимірювання, на яку спрямоване посилання

від атрибуту uom, має бути придатною для коефіцієнту масштабу, як наприклад, процент, проміле або одна мільйонна.

Елемент `gml:semiMinorAxis` містить довжину великої півосі еліпсоїда. Коли включений елемент `gml:isSphere`, еліпсоїд дегенеративний та є сферою. Сфера є цілком визначеною великою напіввіссю, що є радіусом сфери.

#### 12.5.3.6 EllipsoidPropertyType

```
<complexType name="EllipsoidPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Ellipsoid"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:EllipsoidPropertyType` є типом властивості для ролей в асоціації еліпсоїду, що або містить визначення цього еліпсоїду, або посилання на нього.

#### 12.5.3.7 PrimeMeridian, greenwichLongitude

```
<element name="PrimeMeridian" type="gml:PrimeMeridianType"
substitutionGroup="gml:Definition"/>
```

```
<complexType name="PrimeMeridianType">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:greenwichLongitude"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:PrimeMeridian` визначає початок, від якого відраховують значення довготи. Значення початкового меридіану `gml:identifier` за умовчанням є "Greenwich".

```
<element name="greenwichLongitude" type="gml:AngleType"/>
```

`gml:greenwichLongitude` є довготою початкового меридіану, виміряною від меридіану Гринвіча, додатною у східному напрямі. Якщо значення початкового "name" є "Greenwich", то значення `greenwichLongitude` має дорівнювати 0 градусів. У властивості використовують `gml:AngleType`.

#### 12.5.3.8 PrimeMeridianPropertyType, primeMeridianRef

```
<complexType name="PrimeMeridianPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:PrimeMeridian"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:PrimeMeridianPropertyType` є типом властивості для ролей в асоціації до ролей початкового меридіану, що або містить визначення цього меридіану, або посиляється на нього.

## 12.5.4 Інші конкретні дати

### 12.5.4.1 EngineeringDatum

```
<element name="EngineeringDatum" type="gml:EngineeringDatumType"
substitutionGroup="gml:AbstractDatum"/>

<complexType name="EngineeringDatumType">
  <complexContent>
    <extension base="gml:AbstractDatumType"/>
  </complexContent>
</complexType>
```

`gml:EngineeringDatum` визначає початок інженерної референцної системи координат та використовується в області навколо початку цієї системи. Цей початок може бути фіксованим відносно Землі (як визначена точка на будівельному майдані) або може бути визначеною точкою на рухомому об'єкті (як судно чи супутник).

### 12.5.4.2 EngineeringDatumPropertyType

```
<complexType name="EngineeringDatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:EngineeringDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:EngineeringDatumPropertyType` є типом властивості для ролей в асоціації до інженерної дати, що або містить визначення цієї дати, або посиляється на нього.

### 12.5.4.3 ImageDatum

```
<element name="ImageDatum" type="gml:ImageDatumType"
substitutionGroup="gml:AbstractDatum"/>
```

```
<complexType name="ImageDatumType">
  <complexContent>
    <extension base="gml:AbstractDatumType">
      <sequence>
        <element ref="gml:pixelInCell"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:ImageDatum` визначає початок растрової референцної системи координат та використовується тільки в локальному контексті. Для растрової дати вихідним визначенням є звичайно або центр растру, або кут растру. Докладніше див. ISO 19111:2007, В.3.5).

### 12.5.4.4 pixelInCell

```
<element name="pixelInCell" type="gml:CodeWithAuthorityType"/>
```

`gml:pixelInCell` є визначенням способу, в який мережа зображення асоційована з атрибутами дати зображення. Відповідний атрибут `codeSpace` має посилатися на джерело

інформації, яка визначає значення та смисли допустимих текстових значень для цієї властивості.

#### 12.5.4.5 ImageDatumPropertyType

```
<complexType name="ImageDatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:ImageDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:ImageDatumPropertyType є типом властивості для ролей в асоціації до дати зображення, що або містить визначення цієї дати, або посиляється на нього.

#### 12.5.4.6 VerticalDatum

```
<element name="VerticalDatum" type="gml:VerticalDatumType"
substitutionGroup="gml:AbstractDatum"/>
```

```
<complexType name="VerticalDatumType">
  <complexContent>
    <extension base="gml:AbstractDatumType"/>
  </complexContent>
</complexType>
```

gml:VerticalDatum є текстовим описом чи набором параметрів, що ідентифікують окрему поверхню референцного рівня, використану як поверхню нульової висоти, включно з її позицією відносно Землі для будь-якої типу висоти, визнаного цим міжнародним стандартом.

#### 12.5.4.7 VerticalDatumPropertyType

```
<complexType name="VerticalDatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:VerticalDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:VerticalDatumPropertyType є типом властивості для ролей в асоціації до вертикальної дати, що або містить визначення цієї дати, або посиляється на нього.

#### 12.5.4.8 TemporalDatum, origin

```
<element name="TemporalDatum" type="gml:TemporalDatumType"
substitutionGroup="gml:AbstractDatum"/>
```

```
<complexType name="TemporalDatumType">
  <complexContent>
    <extension base="gml:TemporalDatumBaseType">
      <sequence>
        <element ref="gml:origin"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

    </complexContent>
  </complexType>

  <complexType name="TemporalDatumBaseType" abstract="true">
    <complexContent>
      <restriction base="gml:AbstractDatumType">
        <sequence>
          <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:description" minOccurs="0"/>
          <element ref="gml:descriptionReference" minOccurs="0"/>
          <element ref="gml:identifier"/>
          <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:remarks" minOccurs="0"/>
          <element ref="gml:domainOfValidity" minOccurs="0"/>
          <element ref="gml:scope" maxOccurs="unbounded"/>
        </sequence>
        <attribute ref="gml:id" use="required"/>
      </restriction>
    </complexContent>
  </complexType>

```

`gml:TemporalDatum` визначає початок часової референцної системи. Цей тип пропускає елементи `"anchorDefinition"` та `"realizationEpoch"` і додає елемент `"origin"` до типу `dateTime`.

Тип `TemporalDatumBaseType` частково визначає початок часової референцної системи координат. Цей тип обмежує `AbstractDatumType` для видалення елементів `"anchorDefinition"` and `"realizationEpoch"`.

```
<element name="origin" type="dateTime"/>
```

`gml:origin` є початком відліку календарних дат та часу в цій часовій системі.

*Елемент `metaDataProperty` скасований, а елемент `gml:description` частково скасований.*

#### 12.5.4.9 TemporalDatumPropertyType

```

<complexType name="TemporalDatumPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TemporalDatum"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:TemporalDatumPropertyType` є типом властивості для ролей в асоціації до часової дати, що або містить визначення цієї дати, або посилається на нього.

### 12.6 Координатні операції

#### 12.6.1 Огляд

Компоненти схем просторових або часових операцій можна розподілити у п'ять логічних груп, що визначають елементи та типи для XML кодування визначень:

- різних абстрактних координатних операцій;

- різних конкретних типів координатних операцій, включно з трансформаціями та перетвореннями;
- значень та груп абстрактних та конкретних параметрів;
- методів операцій;
- параметрів та груп абстрактних та конкретних операцій.

У компонентах схеми закодовано пакет координатних операцій в моделі UML для ISO 19111, розділ 11.

**Примітка 1.** Схема `coordinateOperations` включає GML документи схеми `coordinateOperations.xsd`. Цей документ схеми ідентифікований таким незалежним від розміщення іменем (з допомогою URN синтаксису):

`urn:x-ogc:specification:gml:schema-xsd:coordinateOperations:3.2.1`

## 12.6.2 Абстрактні координатні операції

### 12.6.2.1 AbstractCoordinateOperation

```
<element name="AbstractCoordinateOperation" type="gml:AbstractCoordinateOperationType"
abstract="true"substitutionGroup="gml:Definition"/>
```

```
<complexType name="AbstractCoordinateOperationType" abstract="true">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:domainOfValidity" minOccurs="0"/>
        <element ref="gml:scope" maxOccurs="unbounded" />
        <element ref="gml:operationVersion" minOccurs="0"/>
        <element ref="gml:coordinateOperationAccuracy" minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:sourceCRS" minOccurs="0"/>
        <element ref="gml:targetCRS" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:AbstractCoordinateOperation` є математичною операцією з координатами, що трансформує або перетворює координати в іншу референцну систему координат. Багато координатних операцій (хоча й не всі), наприклад із РСК А до РСК Б, також унікально визначають зворотну операцію (з РСК Б до РСК А). Іноді алгоритм методу операції для зворотної операції є тим же самим, що і прямий алгоритм, але знаки деяких значень параметрів операції мають бути протилежними. В інших випадках потрібні різні алгоритми для прямих та зворотних операцій, але треба використовувати такі ж самі значення параметрів операції. Якщо потрібні (деякі) цілком відмінні значення параметрів, то треба визначити і відмінні координатні операції.

Необов'язкова властивість `coordinateOperationAccuracy` забезпечує підрахунки впливу цієї координатної операції на точності позицій точок.

### 12.6.2.2 operationVersion

```
<element name="operationVersion" type="string"/>
```

`gml:operationVersion` є версією координатної трансформації (тобто, конкретизацією згідно до стохастичної натури параметрів). Є обов'язковою при описуванні трансформації та не має бути подана для перетворення.

### 12.6.2.3 `coordinateOperationAccuracy`

```
<element name="coordinateOperationAccuracy">
  <complexType>
    <sequence minOccurs="0">
      <element ref="gmd:AbstractDQ_PositionalAccuracy"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</element>
```

`gml:coordinateOperationAccuracy` є роллю в асоціації до об'єкту `DQ_PositionalAccuracy`, як він закодований в ISO/TS 19139, що або містить визначення цієї позиційної точності, або посилається на нього. Цей об'єкт містить оцінку впливу цієї координатної операції на точність позицій. Тобто, він надає оцінку помилки позицій для цільових координат цієї координатної операції з припущенням про відсутність помилок у вихідних координатах.

### 12.6.2.4 `sourceCRS`

```
<element name="sourceCRS" type="gml:CRSPropertyType"/>
```

`gml:sourceCRS` є роллю в асоціації до референцної системи координат, яка є джерелом у цій координатній операції.

### 12.6.2.5 `targetCRS`

```
<element name="targetCRS" type="gml:CRSPropertyType"/>
```

`gml:targetCRS` є роллю в асоціації до цільової референцної системи координат у цій координатній операції.

### 12.6.2.6 `CoordinateOperationPropertyType`, *coordinateOperationRef*

```
<complexType name="CoordinateOperationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractCoordinateOperation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:CoordinateOperationPropertyType` є типом властивості для ролей в асоціації до координатної операції, що або містить визначення цієї координатної операції, або посилається на нього.

### 12.6.2.7 `AbstractSingleOperation`

```
<element name="AbstractSingleOperation" type="gml:AbstractCoordinateOperationType"
  abstract="true" substitutionGroup="gml:AbstractCoordinateOperation"/>
```

`gml:AbstractSingleOperation` є окремою (не зчепленою) координатною операцією.

### 12.6.2.8 `SingleOperationPropertyType`

```
<complexType name="SingleOperationPropertyType">
```

```

<sequence minOccurs="0">
  <element ref="gml:AbstractSingleOperation"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:SingleOperationPropertyType` є типом властивості для ролей в асоціації до одиної операції, що або містить визначення цієї одиної операції, або посилається на нього.

### 12.6.2.9 AbstractGeneralConversion

```

<element name="AbstractGeneralConversion" type="gml:AbstractGeneralConversionType"
abstract="true"substitutionGroup="gml:AbstractOperation"/>

```

```

<complexType name="AbstractGeneralConversionType" abstract="true">
  <complexContent>
    <restriction base="gml:AbstractCoordinateOperationType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:descriptionReference" minOccurs="0"/>
        <element ref="gml:identifier"/>
        <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:remarks" minOccurs="0"/>
        <element ref="gml:domainOfValidity" minOccurs="0"/>
        <element ref="gml:scope" maxOccurs="unbounded"/>
        <element ref="gml:coordinateOperationAccuracy" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="gml:id" use="required"/>
    </restriction>
  </complexContent>
</complexType>

```

`gml:AbstractGeneralConversion` є абстрактною операцією, що не включає зміну геодезичної дати. Найкращим прикладом перетворення координат є картографічна проекція. Параметри, що описують перетворення координат, радше визначені, ніж отримані емпірично. Треба зауважити, що деякі перетворення не мають параметрів. Елементи `gml:operationVersion`, `gml:sourceCRS`, та `gml:targetCRS` не включені в перетворення координат.

Очікується, що абстрактний тип комплексу буде розширений для широко відомих методів операції з багатьма екземплярами перетворення у прикладних схемах, що визначають імена та контенти елементів значень залежно від певних операції та методу. У цьому перетворенні використовують метод операції з асоційованими значеннями параметрів. Однак, методи операції та значення параметрів асоційовані з конкретними підтипами, в не з абстрактним типом. Всі конкретні типи, отримані з цього типу, мають розширювати цей тип, щоб включити елемент `usesMethod`, який посилається на один елемент `OperationMethod`. Аналогічно, всі конкретні типи, отримані з цього типу, мають розширювати цей тип, щоб включити нуль чи більше елементів, які називають



"uses...Value", щоб у кожному використовували тип елемента, яким можна замінити елемент "AbstractGeneralParameterValue".

*Елемент metaDataProperty скасовано, та елемент gml:description частково скасовано.*

#### 12.6.2.10 GeneralConversionPropertyType

```
<complexType name="GeneralConversionPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeneralConversion"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

gml:GeneralConversionPropertyType є типом властивості для ролей в асоціації до загального перетворення, що або містить визначення цього перетворення, або посилається на нього.

#### 12.6.2.11 AbstractGeneralTransformation

```
<element name="AbstractGeneralTransformation"
type="gml:AbstractGeneralTransformationType"
abstract="true"substitutionGroup="gml:AbstractOperation"/>

<complexType name="AbstractGeneralTransformationType" abstract="true">
  <complexContent>
    <restriction base="gml:AbstractCoordinateOperationType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:descriptionReference" minOccurs="0"/>
        <element ref="gml:identifier"/>
        <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:remarks" minOccurs="0"/>
        <element ref="gml:domainOfValidity" minOccurs="0"/>
        <element ref="gml:scope" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:operationVersion"/>
        <element ref="gml:coordinateOperationAccuracy" minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:sourceCRS"/>
        <element ref="gml:targetCRS"/>
      </sequence>
      <attribute ref="gml:id" use="required"/>
    </restriction>
  </complexContent>
</complexType>
```

gml:AbstractGeneralTransformation є абстрактною операцією з координатами, яка звичайно включає зміну дати. Параметри координатної трансформації емпірично отримані з даних, які містять координати набору точок в обох референціальних системах координат. Обчислювальний процес звичайно є “занадто визначеним”, що дозволяє отримання підрахунку помилки (або точності) для трансформації. Також, стохастична

природа параметрів може призводити до чисельних різних версій однієї й тієї ж координатної трансформації. Елементи властивості версій однієї й тієї ж координатної трансформації. Елементи властивості `gml:operationVersion`, `gml:sourceCRS` та `gml:targetCRS` обов'язкові в координатній трансформації.

Припускають, що цей абстрактний комплексний тип можна розширити для добре відомих методів операції з багатьма екземплярами трансформації у прикладних схемах, що визначають імена та контент елементів значень спеціально для окремих методів операції. У цій трансформації використовують метод операції з асоційованими значеннями параметрів. Однак, методи операцій та параметри значень напряду асоційовані з конкретними підтипами, але не з цим абстрактним типом. Всі конкретні типи, отримані з цього типу, мають розширювати цей тип для включення елементу "usesMethod", що посилається на один елемент "OperationMethod". Аналогічно, всі конкретні типи, отримані з цього типу, мають включати один чи більше елементів, кожен із яких називають "uses...Value", щоб у кожному використовували тип елементу, яким можна замінити елемент "AbstractGeneralParameterValue".

*Елемент metaDataProperty скасований, та елемент gml:description частково скасований.*

#### 12.6.2.12 GeneralTransformationPropertyType

```
<complexType name="GeneralTransformationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeneralTransformation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:GeneralTransformationPropertyType` є типом властивості для ролей в асоціації до загальної трансформації, що або містить визначення цієї трансформації, або посилається на нього.

### 12.6.3 Конкретні координатні операції

#### 12.6.3.1 ConcatenatedOperation

```
<element name="ConcatenatedOperation" type="gml:ConcatenatedOperationType"
substitutionGroup="gml:AbstractCoordinateOperation"/>

<complexType name="ConcatenatedOperationType">
  <complexContent>
    <extension base="gml:AbstractCoordinateOperationType">
      <sequence>
        <element ref="gml:coordOperation" minOccurs="2" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

`gml:ConcatenatedOperation` є впорядкованою послідовністю двох чи більше координатних операцій. Ця послідовність операцій обмежена вимогою, згідно до якої вихідна референсна система координат на кроці (n+1) має бути такою ж, що і цільова

референцна система координат на кроці (n). Вихідна референцна система координат на першому кроці та цільова референцна система координат на останньому кроці є референцними системами координат, що є джерелом та ціллю, асоційованими із зчепленою операцією. Замість прямої операції зворотну операцію можна використати для одного чи більше кроків операції, згаданих вище, якщо зворотна операція унікально визначена прямою операцією.

Елементи властивості `gml:coordOperation` є впорядкованою послідовністю асоціацій з двома чи більше операціями, які використовують для цієї пов'язаної операції. `gml:AggregationAttributeGroup` треба використовувати для підтвердження, що асоціації операції `gml:coordOperation` впорядковані.

### 12.6.3.2 coordOperation

```
<element name="coordOperation" type="gml:CoordinateOperationPropertyType"/>
```

`gml:coordOperation` є роллю в асоціації до координатної операції.

### 12.6.3.3 ConcatenatedOperationPropertyType

```
<complexType name="ConcatenatedOperationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:ConcatenatedOperation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:ConcatenatedOperationPropertyType` є типом властивості для ролей в асоціації зчепленої операції, що або містить визначення цієї зчепленої операції, або посиляється на нього.

### 12.6.3.4 PassThroughOperation

```
<element name="PassThroughOperation" type="gml:PassThroughOperationType"
substitutionGroup="gml:AbstractSingleOperation"/>
```

```
<complexType name="PassThroughOperationType">
  <complexContent>
    <extension base="gml:AbstractCoordinateOperationType">
      <sequence>
        <element ref="gml:modifiedCoordinate" maxOccurs="unbounded"/>
        <element ref="gml:coordOperation"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

`gml:PassThroughOperation` встановлює, що підмножина кортежу координат піддається певній прохідній координатній операції.

Елементи властивості `gml:modifiedCoordinate` є впорядкованою послідовністю цілих додатних чисел, що визначає позиції в кортежі координат, на які впливає ця прохідна операція. Треба використовувати `gml:AggregationAttributeGroup` для того, щоб визначити, що елементи `gml:modifiedCoordinate` впорядковані.

### 12.6.3.5 modifiedCoordinate

```
<element name="modifiedCoordinate" type="positiveInteger"/>
```

`gml:modifiedCoordinate` є цілим додатним числом, що визначає позицію в кортежі координат.

### 12.6.3.6 PassThroughOperationPropertyType

```
<complexType name="PassThroughOperationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:PassThroughOperation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:PassThroughOperationPropertyType` є типом властивості для ролей в асоціації до прохідної операції, що або містить визначення цієї прохідної операції, або посилається на нього.

### 12.6.3.7 Conversion

```
<element name="Conversion" type="gml:ConversionType"
substitutionGroup="gml:AbstractGeneralConversion"/>
```

```
<complexType name="ConversionType">
  <complexContent>
    <extension base="gml:AbstractGeneralConversionType">
      <sequence>
        <element ref="gml:method"/>
        <element ref="gml:parameterValue" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:Conversion` є конкретною операцією з координатами, що не передбачає ніяких змін дати. Найбільш відомий приклад перетворення координат – картографічна проекція. Параметри, що описують перетворення координат радше визначені, ніж отримані емпіричною. Слід зауважити, що деякі перетворення не мають параметрів.

Конкретний комплексний тип можна використовувати без використання GML прикладної схеми, яка визначає імена та контенти елементів, спеціалізованих за методом операції, особливо для методів з тільки одним екземпляром перетворення.

### 12.6.3.8 Method

```
<element name="method" type="gml:OperationMethodPropertyType"/>
```

`gml:method` є роллю в асоціації методу операції, що використовують у координатній операції.

### 12.6.3.9 parameterValue

```
<element name="parameterValue" type="gml:AbstractGeneralParameterValuePropertyType"/>
```

`gml:parameterValue` є композитною асоціацією значення параметру або групи значень параметру, що використовують у координатній операції.

#### 12.6.3.10 ConversionPropertyType

```
<complexType name="ConversionPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Conversion"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:ConversionPropertyType` є типом властивості для ролей в асоціації до конкретного перетворення загального призначення, що або містить визначення цього перетворення, або посилається на це визначення.

#### 12.6.3.11 Transformation

```
<element name="Transformation" type="gml:TransformationType"
substitutionGroup="gml:AbstractGeneralTransformation"/>
```

```
<complexType name="TransformationType">
  <complexContent>
    <extension base="gml:AbstractGeneralTransformationType">
      <sequence>
        <element ref="gml:method"/>
        <element ref="gml:parameterValue" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:Transformation` є конкретним елементом об'єкту, отриманого з `gml:GeneralTransformation` (підпункт 12.6.2.11).

Цей конкретний об'єкт можна використовувати для всіх методів операції без використання GML прикладної схеми, що визначає імена та контенти елементів, спеціалізованих за методом операції, особливо для методів з тільки одним екземпляром трансформації.

Елементи `parameterValue` є невпорядкованим списком композитних асоціацій множини значень параметрів, що використовують для цієї операції перетворення.

#### 12.6.3.12 TransformationPropertyType

```
<complexType name="TransformationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Transformation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:TransformationPropertyType` є типом властивості для ролей в асоціації до трансформації, що або містить визначення цієї трансформації, або посилається на нього.

## 12.6.4 Значення параметрів та групи

### 12.6.4.1 AbstractGeneralParameterValue

```
<element name="AbstractGeneralParameterValue" type="gml:AbstractGeneralParameterValue" abstract="true"
substitutionGroup="gml:AbstractObject"/>
```

```
<complexType name="AbstractGeneralParameterValue" abstract="true">
  <sequence/>
</complexType>
```

`gml:AbstractGeneralParameterValue` є значенням абстрактних параметру чи групою значень параметру.

Очікують, що цей абстрактний комплексний буде розширений та обмежений для широко відомих методів операції з багатьма екземплярами у прикладних схемах, що визначають імена та контенти елементів, спеціалізовані за методом операції. Специфічні елементи значення параметру напряду містяться в конкретних підтипах, а не в цьому абстрактному типі. Всі конкретні типи, отримані з цього типу, мають розширювати цей тип для включення одного елементу "...Value" з відповідним типом, що має бути одним із типів елементів, дозволених у `ParameterValue`. Крім того, всі похідні конкретні типи мають розширювати цей тип для включення елементу властивості "operationParameter", що посилається на один елемент, яким можна замінити елемент об'єкту "OperationParameter".

### 12.6.4.2 AbstractGeneralParameterValuePropertyType

```
<complexType name="AbstractGeneralParameterValuePropertyType">
  <sequence>
    <element ref="gml:AbstractGeneralParameterValue" />
  </sequence>
</complexType>
```

`gml:AbstractGeneralParameterValuePropertyType` є типом властивості для включених ролей в асоціації до значення параметру або групи значень параметру, що завжди містить ці значення.

### 12.6.4.3 ParameterValue

```
<element name="ParameterValue" type="gml:ParameterValue"
substitutionGroup="gml:AbstractGeneralParameterValue"/>
```

```
<complexType name="ParameterValue">
  <complexContent>
    <extension base="gml:AbstractGeneralParameterValue">
      <sequence>
        <choice>
          <element ref="gml:value"/>
          <element ref="gml:dmsAngleValue"/>
          <element ref="gml:stringValue"/>
          <element ref="gml:integerValue"/>
          <element ref="gml:booleanValue"/>
          <element ref="gml:valueList"/>
          <element ref="gml:integerValueList"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        <element ref="gml:valueFile"/>
    </choice>
    <element ref="gml:operationParameter"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

`gml:ParameterValue` є значенням параметру та впорядкованою послідовністю значень або посиланням на файл значень параметру. Цей конкретний комплексний тип можна використовувати для методів операції без використання прикладної схеми, яка визначає імена та контенти елементів, спеціалізовані за методами операцій, особливо для методів з тільки одним екземпляром. Цей конкретний комплексний тип можна використовувати, розширювати або обмежувати для широко відомих методів операції, особливо для методів з багатьма екземплярами.

*Елемент `dmsAngleValue` скасований.*

#### **12.6.4.4 value**

```
<element name="value" type="gml:MeasureType"/>
```

`gml:value` є числовим значенням параметру операції з її асоційованою одиницею вимірювання.

#### **12.6.4.5 stringValue**

```
<element name="stringValue" type="string"/>
```

`gml:stringValue` є значенням параметру операції у вигляді символного рядка. Воно не має асоційованої одиниці вимірювання.

#### **12.6.4.6 integerValue**

```
<element name="integerValue" type="positiveInteger"/>
```

`gml:integerValue` є цілим додатним значенням параметру операції, що звичайно використовують у рахунку. Ціле значення не має асоційованої одиниці вимірювання.

#### **12.6.4.7 booleanValue**

```
<element name="booleanValue" type="boolean"/>
```

`gml:booleanValue` є булевським значенням параметру операції. Булевське значення не має асоційованої одиниці вимірювання.

#### **12.6.4.8 valueList**

```
<element name="valueList" type="gml:MeasureListType"/>
```

`gml:valueList` є впорядкованою послідовністю двох чи більше числових значень списку параметрів операції, де кожне значення має ту ж саму асоційовану одиницю вимірювання. Елемент цього типу містить послідовність довгих десяткових значень, розділених пробілами.

#### **12.6.4.9 integerValueList**

```
<element name="integerValueList" type="gml:integerList"/>
```

`gml:integerValueList` є впорядкованою послідовністю двох чи більше цілочислових значень списку параметрів операції, що звичайно використовують для рахунку. Ці

цілочислові значення не мають асоційованої одиниці вимірювання. Елемент цього типу містить послідовність цілочислових значень, розділених пробілами.

#### 12.6.4.10 valueFile

```
<element name="valueFile" type="anyURI"/>
```

gml:valueFile є посиланням на файл або частину файлу, що містить одне чи більше значень параметру, кожне з яких є числовим значенням з асоційованою одиницею вимірювання. При посиланні на частину файлу цей файл має містити кілька ідентифікованих частин, таких як XML закодований документ. Крім цього, файл чи частина файлу, на які спрямоване посилання, можуть посилатися на іншу частину того ж або іншого файлу так, як це дозволено в XML документах.

#### 12.6.4.11 operationParameter

```
<element name="operationParameter" type="gml:OperationParameterPropertyType"/>
```

gml:operationParameter є роллю в асоціації до параметру операції, для якого це є значенням.

#### 12.6.4.12 ParameterValueGroup

```
<element name="ParameterValueGroup" type="gml:ParameterValueGroupType"
substitutionGroup="gml:AbstractGeneralParameterValue"/>
```

```
<complexType name="ParameterValueGroupType">
  <complexContent>
    <extension base="gml:AbstractGeneralParameterValue">
      <sequence>
        <element ref="gml:parameterValue" minOccurs="2" maxOccurs="unbounded"/>
        <element ref="gml:group"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

gml:ParameterValueGroup є групою пов'язаних значень параметру. Така ж група може зустрічатися більш ніж один раз у перетворенні, трансформації або групі значень параметру ParameterValueGroup вищого рівня, якщо ці екземпляри містять інші типи одного чи більше значень параметру parameterValues, що можна легко відрізнити серед цих груп. Цей конкретний комплексний тип можна використовувати для методів операції без допомоги прикладної схеми, що визначає імена та контенти елементів, спеціалізовані за методом операції. Цей комплексний тип можна використовувати, розширювати або обмежувати для широко відомих методів операції, особливо для методів із тільки одним екземпляром.

Елементи parameterValue є невпорядкованою множиною ролей в асоціації композиції до значень параметру та груп значень, включених у цю групу.

#### 12.6.4.13 group

```
<element name="group" type="gml:OperationParameterGroupPropertyType"/>
```

gml:group є роллю в асоціації до групи параметрів операції, для якої цей елемент надає значення параметрів.



## 12.6.5 Метод операції

### 12.6.5.1 OperationMethod

```
<element name="OperationMethod" type="gml:OperationMethodType" substitutionGroup="gml:Definition"/>

<complexType name="OperationMethodType">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <choice>
          <element ref="gml:formulaCitation"/>
          <element ref="gml:formula"/>
        </choice>
        <element ref="gml:sourceDimensions" minOccurs="0"/>
        <element ref="gml:targetDimensions" minOccurs="0"/>
        <element ref="gml:generalOperationParameter" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:OperationMethod` є методом (алгоритмом чи процедурою), що використовують для здійснення координатної операції. У більшості методів операції використовують кілька параметрів операції, хоча в деяких координатних перетвореннях їх не використовують. Кожна координатна операція з використанням методу надає значення цим параметрам.

Елементи `gml:parameter` є невпорядкованим списком асоціацій до набору параметрів операції та групи параметрів, використаних у цьому методі операції.

#### 12.6.5.2 formula

```
<element name="formula" type="gml:CodeType"/>
```

`gml:formula` є формулою чи формулами або процедурою, використаними в цьому методі операції.

Використання атрибуту `codespace` скасовано. Значення властивості має бути символьним рядком.

```
<element name="formulaCitation">
  <complexType>
    <sequence minOccurs="0">
      <element ref="gmd:CI_Citation"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</element>
```

`gml:formulaCitation` надає посилання на публікацію, в якій подані формула чи формули або процедура, використані в методі координатної операції.

#### 12.6.5.3 sourceDimensions

```
<element name="sourceDimensions" type="positiveInteger"/>
```

`gml:sourceDimensions` є числом вимірів у вихідній референційній системі координат у цьому методі операції.

#### 12.6.5.4 `targetDimensions`

```
<element name="targetDimensions" type="positiveInteger"/>
```

`gml:targetDimensions` є числом вимірів у цільовій референційній системі координат у цьому методі операції.

#### 12.6.5.5 `parameter`

```
<element name="parameter" type="gml:AbstractGeneralOperationParameterPropertyType"/>
```

`gml:parameter` є асоціацією до параметру операції чи групи параметрів.

#### 12.6.5.6 `OperationMethodPropertyType`

```
<complexType name="OperationMethodPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:OperationMethod"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:OperationMethodPropertyType` є типом властивості для ролей в асоціації конкретного методу операції загального призначення, що або містить визначення цього методу, або посилається на нього.

### 12.6.6 Параметри та групи операцій

#### 12.6.6.1 `generalOperationParameter`

```
<element name="AbstractGeneralOperationParameter"
type="gml:AbstractGeneralOperationParameterType"
abstract="true" substitutionGroup="gml:Definition"/>
```

```
<complexType name="AbstractGeneralOperationParameterType" abstract="true">
  <complexContent>
    <extension base="gml:IdentifiedObjectType">
      <sequence>
        <element ref="gml:minimumOccurs" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:GeneralOperationParameter` є абстрактним визначенням параметру чи групи параметрів, які використовують у методі операції.

#### 12.6.6.2 `minimumOccurs`

```
<element name="minimumOccurs" type="nonNegativeInteger"/>
```

`ml:minimumOccurs` є мінімальним числом разів, коли потребувалися значення для цієї групи параметрів чи параметру. Якщо цей атрибут пропущений, то це мінімальне число має становити один.

### 12.6.6.3 AbstractGeneralOperationParameterPropertyType

```
<complexType name="AbstractGeneralOperationParameterPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractGeneralOperationParameter"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:AbstractGeneralOperationParameterPropertyType` є типом властивості для ролей в асоціації до параметру операції чи групи, що або містить визначення цього параметру або групи, або посилається на нього.

### 12.6.6.4 OperationParameter

```
<element name="OperationParameter" type="gml:OperationParameterType"
substitutionGroup="gml:AbstractGeneralOperationParameter"/>
```

```
<complexType name="OperationParameterType">
  <complexContent>
    <extension base="gml:AbstractGeneralOperationParameterType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
```

`gml:OperationParameter` є визначенням параметру, що використовують у методі операції. Більшість значень параметрів числові, але також можливі інші типи значень параметрів. Очікують, що цей комплексний тип має бути використаний або розширений для всіх методів операції без визначення імен елементів, спеціалізованих за методом операції.

### 12.6.6.5 OperationParameterPropertyType

```
<complexType name="OperationParameterPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:OperationParameter"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

`gml:OperationParameterPropertyType` є типом властивості для ролей в асоціації до параметру операції, що або містить визначення цього параметру, або посилається на нього.

### 12.6.6.6 OperationParameterGroup

```
<element name="OperationParameterGroup" type="gml:OperationParameterGroupType"
substitutionGroup="gml:AbstractGeneralOperationParameter"/>
```

```
<complexType name="OperationParameterGroupType">
  <complexContent>
    <extension base="gml:AbstractGeneralOperationParameterType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
```

```

    <element ref="gml:maximumOccurs" minOccurs="0"/>
    <element ref="gml:generalOperationParameter" minOccurs="2" maxOccurs="unbounded"/>
  </sequence>
</extension>
</complexContent>
</complexType>

```

`gml:OperationParameterGroup` є визначенням групи параметрів, використаних у методі операції. Очікують, що цей комплексний тип можна розширювати для всіх застосовних методів операції без визначення імен елементів, спеціалізованих за методом операції.

Елементи `generalOperationParameter` є неупорядкованим списком асоціацій до набору параметрів операції, що є членами цієї групи.

#### 12.6.6.7 maximumOccurs

```
<element name="maximumOccurs" type="positiveInteger"/>
```

`gml:maximumOccurs` є максимальним числом разів, коли можуть бути включені значення для цієї групи параметрів. Якщо цей атрибут пропущений, то максимальне число має бути один.

#### 12.6.6.8 OperationParameterPropertyType

```

<complexType name="OperationParameterGroupPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:OperationParameterGroup"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:OperationParameterPropertyType` є типом властивості для ролей в асоціації до групи параметрів операцій, що або містить визначення цієї групи параметрів, або посилається на нього.

## 13 GML СХЕМА – ТОПОЛОГІЯ

### 13.1 Загальні концепції

#### 13.1.1 Огляд

Топологія є розділом математики, що описує властивості об'єктів, які є інваріантними за безперервної деформації. Наприклад, коло є топологічно еквівалентним еліпсу, оскільки одне може бути трансформоване в друге шляхом розтягування. У географічному моделюванні основне використання топології полягає в прискоренні обчислювальної геометрії. Конструкти топології дозволяють характеризувати просторові відношення між об'єктами з використанням простих комбінаторних або алгебраїчних алгоритмів. Топологія, здійснена на базі відповідної геометрії, також забезпечує компактний та однозначний механізм для подання спільної геометрії між географічними об'єктами.

**Примітка 1.** Модель топології GML відповідає ISO 19107. Відповідні базові концепції типів та елементів моделі топології GML більш докладно розглянуті в цьому документі.

У цьому розділі описані компоненти схеми топології, як визначено у GML.

**Примітка 2.** Відповідний документ схеми топології, topology.xsd (див. додаток В), визначений таким незалежним від розміщення іменем (з використанням URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:topology:3.2.1

Існують чотири класи (що можуть мати екземпляри) топологічних примітивів, один для кожного виміру в трьохвимірній системі. Крім того, також підтримуються топологічні комплекси.

Є строга симетрія в відносинах (топологічна межа та співмежа) між топологічними елементами у суміжних розмірностях. Топологічні примітиви обмежені спрямованими примітивами нижчої розмірності. Співмежа кожного топологічного примітиву сформована із спрямованих топологічних примітивів розмірності, вищої на одну ступінь.

### 13.1.2 Зв'язок з ISO 19107

Компоненти просторової топології GML схеми, визначені в цьому розділі, забезпечують узгоджене часткове впровадження просторової схеми (топології) стандарту ISO 19107. Цей зв'язок докладно розглянуте в Г.2.3.

Типи топології стандарту ISO 19107, впроваджені в GML, визначені в ISO 19107; деякі додаткові обмеження визначені для тих типів в ISO 19107, які є також обмеженнями на компоненти просторової топології GML схеми.

Крім цього, GML визначає компліментарні компоненти схеми просторової топології, як визначено в Г.3.11.

## 13.2 Абстрактна топологія

```
<complexType name="AbstractTopologyType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType"/>
  </complexContent>
</complexType>
```

```
<element name="AbstractTopology" type="gml:AbstractTopologyType" abstract="true"
substitutionGroup="gml:AbstractGML"/>
```

Цей абстрактний тип забезпечує кореневий або базовий тип для всіх топологічних елементів, включно з примітивами та комплексами. Він успадковує AbstractGMLType, тому може бути ідентифікований з допомогою атрибуту gml:id.

gml:AbstractTopology реалізує TP\_Object стандарту ISO 19107 (див. Г.2.4.2 та ISO 19107:2003, пункт 7.2.2).

### 13.3 Топологічні примітиви

#### 13.3.1 Абстрактні топологічні примітиви

##### 13.3.1.1 AbstractTopoPrimitive, AbstractTopoPrimitive

```
<complexType name="AbstractTopoPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="AbstractTopoPrimitive" type="gml:AbstractTopoPrimitiveType" abstract="true"
substitutionGroup="gml:AbstractTopology"/>
```

gml:AbstractTopoPrimitive реалізує TP\_Primitive стандарту ISO 19107 (див. Г.2.4.3 та ISO 19107:2003, пункт 7.3.10). Цей абстрактний тип діє як базовий тип для всіх топологічних примітивів. Топологічні примітиви є атомарними (найменшими можливими) одиницями комплексу топології.

Кожний топологічний примітив може містити посилання на інші топологічні примітиви співвиміру 2 чи більше (gml:isolated).

*Приклад 1.* Грані можуть містити ізольовані вузли, а тіла можуть містити ізольовані вузли та ребра.

Навпаки, вузли можуть мати грані як вмістилища, а вузли та ребра можуть мати тіла solid як вмістилища (gml:container, що реалізовано в підтипах).

### 13.3.2 Топологічні примітиви (нульвимірні)

#### 13.3.2.1 NodeType, Node

```
<complexType name="NodeType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element name="container" type="gml:FaceOrTopoSolidPropertyType" minOccurs="0"/>
        <element ref="gml:directedEdge" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:pointProperty" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Node" type="gml:NodeType" substitutionGroup="gml:AbstractTopoPrimitive"/>
```

gml:Node являє нульвимірний примітив та реалізує TP\_Node стандарту ISO 19107 (див. Г.2.4.3 та ISO 19107:2003, підпункт 7.3.12).

Елемент властивості gml:container реалізує роль того ж самого імені з асоціації “Isolated In” стандарту ISO 19107 (див. ISO 19107:2003, підпункт 7.3.10.4 та Г.2.4.3).

```
<complexType name="FaceOrTopoSolidPropertyType">
  <choice minOccurs="0">
    <element ref="gml:Face"/>
    <element ref="gml:TopoSolid"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Необов’язкова співмежа вузлу (gml:directedEdge) є послідовністю спрямованих ребер, що сходяться в цьому вузлі. Ребра, що виходять із цього вузла, мають у суміжній межі від’ємну орієнтацію.

Атрибут aggregationType, якщо він поданий, повинен мати значення “sequence”.

Вузол можна реалізувати через нульвимірний геометричний примітив (gml:pointProperty).

### 13.3.2.2 DirectedNodeType, directedNode

```
<element name="directedNode" type="gml:DirectedNodeType"/>
```

```
<complexType name="DirectedNodeType">
  <sequence minOccurs="0">
    <element ref="gml:Node"/>
  </sequence>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Елемент властивості gml:directedNode описує межу топологічних ребер та використовується в підтримці топологічних точкових об'єктів через вираз gml:TopoPoint (див. далі). Атрибут орієнтації типу gml:SignType виражає смисл, у якому використовують включений елемент: початковий (“-”) або кінцевий (“+”) вузол.

### 13.3.3 Топологічні примітиви (одновимірні)

#### 13.3.3.1 EdgeType, Edge

```
<complexType name="EdgeType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element name="container" type="gml:TopoSolidPropertyType" minOccurs="0"/>
        <element ref="gml:directedNode" minOccurs="2" maxOccurs="2"/>
        <element ref="gml:directedFace" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:curveProperty" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Edge" type="gml:EdgeType" substitutionGroup="gml:AbstractTopoPrimitive"/>
```

gml:Edge являє одновимірний примітив та реалізує TP\_Edge стандарту ISO 19107 (див. Г.2.4.3 та ISO 19107:2003, пункт 7.3.14).

Елемент властивості gml:container реалізує роль того ж самого імені з асоціації “Isolated In” стандарту ISO 19107 (див. ISO 19107:2003, підпункт 7.3.10.4 та Г.2.4.3).

Топологічна межа ребра (gml:directedNode) складається з від'ємно спрямованого початкового вузлу та додатньо спрямованого кінцевого вузлу.

Необов'язкова співмежа ребра (gml:directedFace) є круговою послідовністю спрямованих граней, що сходяться на цьому ребрі, у порядку документу. У двохвимірному випадку орієнтація грані ліворуч від ребра є "+"; орієнтація грані праворуч від її ребра є "-".

Якщо поданий атрибут aggregationType, то він повинен мати значення “sequence”.

Ребро можна (необов'язково) реалізувати через одновимірний геометричний примітив (gml:curveProperty).

### 13.3.3.2 DirectedEdgePropertyType, directedEdge

```
<element name="directedEdge" type="gml:DirectedEdgePropertyType"/>

<complexType name="DirectedEdgePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Edge"/>
  </sequence>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Елемент властивості gml:directedEdge описує межу граней топології, співмежу вузлів топології, та може бути використаний у підтримці топологічних лінійних об'єктів через вираз gml:TopoCurve (див. далі). Атрибут орієнтації типу gml:SignType виражає смисл, у якому використовують включене ребро, тобто прямий чи зворотний напрямок.

### 13.3.4 Топологічні примітиви (двохвимірні)

#### 13.3.4.1 FaceType, Face

```
<complexType name="FaceType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element name="isolated" type="gml:NodePropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:directedEdge" maxOccurs="unbounded"/>
        <element ref="gml:directedTopoSolid" minOccurs="0" maxOccurs="2"/>
        <element ref="gml:surfaceProperty" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
      <attribute name="universal" type="boolean" use="optional" default="false"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Face" type="gml:FaceType" substitutionGroup="gml:AbstractTopoPrimitive"/>
```

gml:Face являє двухвимірний примітив топології та реалізує TP\_Face стандарту ISO 19107 (див. Г.2.4.3 та and ISO 19107:2003, пункт 7.3.16).

Елемент властивості gml:isolated реалізує роль того ж самого імені з асоціації “Isolated In” стандарту ISO 19107 (див. ISO 19107:2003, підпункт 7.3.10.4 та Г.2.4.3).

Топологічна межа грані (gml:directedEdge) складається з послідовності спрямованих ребер. Якщо поданий атрибут aggregationType, то він повинен мати значення “sequence”.

**Примітка 1.** Всі ребра, асоційовані з гранню, включно із висячими ребрами, знаходяться на межі. Висяче ребро має одну й ту ж грань з обох боків. Тому висяче ребро має два різних вузли на своїй межі. Висяче ребро може мати нуль, один або два вузли на межі з іншими ребрами на межі грані. Два елементи directedEdge з протилежними орієнтаціями посилаються на



кожне висяче ребро на межі грані. Невисячі ребра на межі грані містять одне чи більше топологічних кілець. Кожне таке кільце складається з спрямованих граней `directedEdge`, з'єднаних у цикл та орієнтованих із гранню, яка знаходиться ліворуч.

Необов'язкова співмежа грані (`gml:directedTopoSolid`) є парою спрямованих тіл, що обмежені цією гранню. Додатньо спрямоване тіло відповідає тілу, що лежить у напрямку від'ємно спрямованого перпендикуляру до грані в будь-якій геометричній реалізації.

Грань можна (необов'язково) реалізувати через двохвимірний геометричний примітив `primitive` (`gml:surfaceProperty`).

Якщо топологічне подання існує як безмежна множина (наприклад, евклідова площина), то `gml:Face` має вказувати, чи є воно універсальною гранню, щоб забезпечити цілісність топологічного подання. Щоб вказувати це, використовують необов'язковий універсальний атрибут булевського типу.

**Примітка 2.** Універсальна поверхня звичайно не є частиною жодного об'єкту та використовується для подання необмеженої частини набору даних. Її внутрішню межу (він не має зовнішньої межі) звичайно розглядали б як зовнішню межу карти, поданої набором даних.

#### 13.3.4.2 `DirectedFacePropertyType`, `directedFace`

```
<element name="directedFace" type="gml:DirectedFacePropertyType"/>
```

```
<complexType name="DirectedFacePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Face"/>
  </sequence>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Елемент властивості `gml:directedFace` описує межу тіл топології, у співмежі ребер топології та використовується в підтримці географічних об'єктів поверхні через вираз `gml:TopoSurface` (див. далі). Атрибут орієнтації типу `gml:SignType` виражає смисл, у якому використовують включену грань, тобто всередину чи назовні відносно перпендикуляру на поверхню в будь-якій геометричній реалізації.

### 13.3.5 Топологічні примітиви (трьохвимірні)

#### 13.3.5.1 `TopoSolidType`, `TopoSolid`

```
<complexType name="TopoSolidType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element name="isolated" type="gml:NodeOrEdgePropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:directedFace" maxOccurs="unbounded"/>
        <element ref="gml:solidProperty" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
      <attribute name="universal" type="boolean" use="optional" default="false"/>
    </extension>
  </complexContent>
```

```
</complexType>
```

```
<element name="TopoSolid" type="gml:TopoSolidType" substitutionGroup="gml:AbstractTopoPrimitive"/>
```

gml:TopoSolid являє примітив трьохвимірної топології та реалізує TP\_Solid стандарту ISO 19107 (див. Г.2.4.3 та ISO 19107:2003, пункт 7.3.18).

Елемент властивості gml:isolated реалізує роль того ж самого імені з асоціації “Isolated In” стандарту ISO 19107 (див. ISO 19107:2003, підпункт 7.3.10.4 та Г.2.4.3).

```
<complexType name="NodeOrEdgePropertyType">
  <choice minOccurs="0">
    <element ref="gml:Node"/>
    <element ref="gml:Edge"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Топологічна межа тіла (gml:directedFace) складається з набору спрямованих граней.

**Примітка 1.** Всі грані, асоційовані з тілом, включно з висячими гранями, знаходяться на межі. Висяча грань має одне й те ж тіло з обох боків. Два елементи directedFace з протилежними орієнтаціями посилаються на кожну висячу грань у межі топологічного тіла.

Тіло може (необов'язково) бути реалізоване через трьохвимірний геометричний примітив (gml:solidProperty).

gml:TopoSolid має вказувати, чи це є універсальне тіло, щоб забезпечити цілісність подання топології. Щоб вказувати це, використовують необов'язковий універсальний атрибут булевського типу, значенням якого за умовчанням буде false (хибність).

**Примітка 2.** Універсальне топологічне тіло звичайно не є частиною жодного об'єкту та використовується для подання необмеженої частини набору даних. Його внутрішню межу (він не має зовнішньої межі) звичайно розглядали б як зовнішню межу набору даних.

### 13.3.5.2 DirectedTopoSolidPropertyType, directedTopoSolid

```
<element name="directedTopoSolid" type="gml:DirectedTopoSolidPropertyType"/>
```

```
<complexType name="DirectedTopoSolidPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TopoSolid"/>
  </sequence>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Елемент властивості gml:directedSolid описує співмежу граней топології та використовується в підтримці об'ємних об'єктів через вираз gml:TopoVolume (див. далі). Атрибут орієнтації типу gml:SignType виражає смисл, у якому включене тіло зустрічається на співмежі грані. У контексті виразу gml:TopoVolume атрибут орієнтації не має смислу.

## 13.4 Топологічні колекції

### 13.4.1 Топологічна колекція (нульвимірна)

#### 13.4.1.1 TopoPointType, gml:TopoPoint

```
<complexType name="TopoPointType">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:directedNode"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="TopoPoint" type="gml:TopoPointType"/>
```

Цільовим використанням gml:TopoPoint є присутність в точковому географічному об'єкті для подання структурних і, можливо, геометричних відношень цього об'єкту до інших географічних об'єктів через визначення спільних вузлів.

**Примітка 1.** Орієнтація orientation, призначена спрямованому вузлу gml:directedNode, не має смислу в цьому контексті. Вона зберігається для симетрії з відповідними елементами інших розмірностей (див. далі).

#### 13.4.1.2 TopoPointPropertyType, topoPointProperty

```
<complexType name="TopoPointPropertyType">
  <sequence>
    <element ref="gml:TopoPoint"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="topoPointProperty" type="gml:TopoPointPropertyType"/>
```

Елемент властивості gml:topoPointProperty можна використовувати в географічних об'єктах для подання їх відношень до вузлу топології, на який спрямоване посилання.

### 13.4.2 Топологічна колекція (одновимірна)

#### 13.4.2.1 TopoCurveType, TopoCurve

```
<complexType name="TopoCurveType">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:directedEdge" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="TopoCurve" type="gml:TopoCurveType"/>
```

`gml:TopoCurve` являє однорідний топологічний вираз, послідовність спрямованих ребер, які, якщо реалізовані, є ізоморфними до геометричного криволінійного примітиву. Цільове використання `gml:TopoCurve` полягає в тому, що він має бути в лінійному географічному об'єкті для подання геометричних відношень цього об'єкту до інших географічних об'єктів через визначення спільних ребер.

Атрибут `aggregationType`, якщо поданий, повинен мати значення "sequence".

#### **13.4.2 TopoCurvePropertyType, topoCurveProperty**

```
<complexType name="TopoCurvePropertyType">
  <sequence>
    <element ref="gml:TopoCurve"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

```
<element name="topoCurveProperty" type="gml:TopoCurvePropertyType"/>
```

Елемент властивості `gml:topoCurveProperty` можна використовувати в географічних об'єктах для подання їх відношень до ребер топології, на які спрямовані посилання.

#### **13.4.3 Топологічна колекція (двохвимірна)**

##### **13.4.3.1 TopoSurfaceType, TopoSurface**

```
<complexType name="TopoSurfaceType">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:directedFace" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="TopoSurface" type="gml:TopoSurfaceType"/>
```

`gml:TopoSurface` являє однорідний топологічний вираз, набір спрямованих граней, які, якщо реалізовані, є ізоморфними до геометричного примітиву поверхні. Цільове використання `gml:TopoSurface` полягає в тому, що він має бути в географічному об'єкті поверхня для подання структурних і, можливо, геометричних відношень цього об'єкту поверхні до інших географічних об'єктів через визначення спільних граней.

##### **13.4.3.2 TopoSurfacePropertyType, topoSurfaceProperty**

```
<complexType name="TopoSurfacePropertyType">
  <sequence>
    <element ref="gml:TopoSurface"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

```
<element name="topoSurfaceProperty" type="gml:TopoSurfacePropertyType"/>
```

Елемент властивості `gml:topoSurfaceProperty` можна використати в географічних об'єктах для вираження їх відношення до граней топології, на які спрямовані посилання.

### 13.4.4 Топологічна колекція (трьохвимірна)

#### 13.4.4.1 TopoVolumeType, TopoVolume

```
<complexType name="TopoVolumeType">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:directedTopoSolid" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="TopoVolume" type="gml:TopoVolumeType"/>
```

`gml:TopoVolume` являє однорідний топологічний вираз, набір спрямованих топологічних тіл, які, якщо реалізовані, є ізоморфними геометричному примітиву тіло. Цільове використання `gml:TopoVolume` полягає в тому, що він має бути в географічному об'єкті тіло для подання структурних та геометричних відношень цього об'єкту до інших географічних об'єктів через визначення спільних тіл.

**Примітка 1.** Орієнтація `orientation`, призначена для `gml:directedSolid`, не має смислу в трьох вимірах. Вона зберігається для симетрії з відповідними елементами інших розмірностей (див. далі).

#### 13.4.4.2 TopoVolumePropertyType, topoVolumeProperty

```
<complexType name="TopoVolumePropertyType">
  <sequence>
    <element ref="gml:TopoVolume"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

```
<element name="topoVolumeProperty" type="gml:TopoVolumePropertyType"/>
```

Елемент `gml:topoVolumeProperty` можна використовувати в географічних об'єктах для вираження їх відношень до топологічного об'єму, на який спрямоване посилання.

### 13.5 Топологічний комплекс

#### 13.5.1 TopoComplexType, TopoComplex

```
<complexType name="TopoComplexType">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:maximalComplex"/>
        <element ref="gml:superComplex" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

    <element ref="gml:subComplex" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="gml:topoPrimitiveMember" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="gml:topoPrimitiveMembers" minOccurs="0"/>
  </sequence>
  <attribute name="isMaximal" type="boolean" default="false"/>
  <attributeGroup ref="gml:AggregationAttributeGroup"/>
</extension>
</complexContent>
</complexType>

```

```

<element name="TopoComplex" type="gml:TopoComplexType"
substitutionGroup="gml:AbstractTopology"/>

```

gml:TopoComplex є колекцією топологічних примітивів та реалізує TP\_Complex стандарту ISO 19107 (див. Г.2.4.4 та ISO 19107:2003, пункт 7.4.2).

Кожний комплекс містить посилання на свій максимальний комплекс (gml:maximalComplex) та (необов'язково) на підкомплекси та надкомплекси (gml:subComplex, gml:superComplex).

Топологічний комплекс містить свої примітивні та підкомплексні члени.

**Примітка 1.** Максимальний комплекс є комплексом, що не входить до надкомплексу. Є один і тільки один максимальний комплекс для топологічного manifold.

### **13.5.2 Максимальний комплекс, підкомплекси (sub-complexes), надкомплекси (super-complexes)**

```

<element name="subComplex" type="gml:TopoComplexMemberType"/>
<element name="superComplex" type="gml:TopoComplexMemberType"/>
<element name="maximalComplex" type="gml:TopoComplexMemberType"/>

```

Елементи властивості gml:subComplex, gml:superComplex та gml:maximalComplex надають кодування для відношень між топологічними комплексами, як описано вище для gml:TopoComplex.

### **13.5.3 topoPrimitiveMember**

```

<element name="topoPrimitiveMember" type="gml:TopoPrimitiveMemberType"/>

```

```

<complexType name="TopoPrimitiveMemberType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractTopoPrimitive" />
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Елемент властивості gml:topoPrimitiveMember кодує відношення між комплексом топології та окремим примітивом топології.

### **13.5.4 topoPrimitiveMembers**

```

<element name="topoPrimitiveMembers" type="gml:TopoPrimitiveArrayAssociationType"/>

```

```

<complexType name="TopoPrimitiveArrayAssociationType">
  <sequence minOccurs="0" maxOccurs="unbounded">

```

```

    <element ref="gml:AbstractTopoPrimitive" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Елемент властивості `gml:topoPrimitiveMembers` кодує відношення між комплексом топології та довільним числом примітивів топології.

**Примітка 1.** Оскільки значенням властивості може бути кілька топологічних примітивів, елементи, якими подають примітиви топології, завжди закодовані як включені.

### 13.5.5 TopoComplexPropertyType, topoComplexProperty

```

<complexType name="TopoComplexMemberType">
  <sequence minOccurs="0">
    <element ref="gml:TopoComplex" />
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

```

<element name="topoComplexProperty" type="gml:TopoComplexMemberType"/>

```

Елемент властивості `gml:topoComplexProperty` кодує відношення між GML об'єктом та комплексом топології.

*Приклад 1.* Це дозволяє колекції географічних об'єктів містити комплекс топології, який містить об'єкти топології, на які посилаються члени колекції географічних об'єктів.

## 14 GML СХЕМА – ЧАСОВА ІНФОРМАЦІЯ ТА ДИНАМІЧНІ ОБ'ЄКТИ

### 14.1 Загальні концепції

#### 14.1.1 Загальний огляд

GML часові схеми включають компоненти для опису часової геометрії та топології, часових референціальних систем і часових характеристик географічних даних. Модель, що лежить в основі подання, складає профіль концептуальної схеми, описаної в ISO 19108. Основна просторово-часова модель призначена для пристосування фіксації часу до рівнів географічного об'єкту та атрибуту; базова підтримка для стеження за рухомими географічними об'єктами також включена.

Час вимірюють по двох типах шкали: інтервальної та порядковій. Інтервальна шкала підтримує основу для вимірювання тривалості, а порядкова шкала забезпечує інформацію тільки про відносне розташування в часі.

*Приклад 1.* Стратиграфічна послідовність геологічної шкали часу.

Інші два Міжнародних стандарти ISO, доречні в описі часових об'єктів: ISO 8601, що описує кодування моментів у часі та часових періодів як текстових рядків з особливою структурою та пунктуацією; ISO 11404, що забезпечує детальний опис інтервалів часу як частину загального обговорення типів даних, незалежних від мови.

Часові схеми покривають дві взаємопов'язані теми та забезпечують компоненти базової схеми для подання часових моментів та періодів, часової топології та референціальних систем; компоненти більш спеціалізованих схем, які визначають компоненти, що використовуються для динамічних географічних об'єктів. Екземпляри часових геометричних типів використовують як значення для часових властивостей географічних об'єктів.

**Примітка 1.** Головний документ часової схеми визначений таким незалежним від розташування іменем (з допомогою URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:temporal:3.2.1

Документ схеми часової топології визначений таким незалежним від розташування іменем (з допомогою URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:temporalTopology:3.2.1

Документ схеми для часових референцних систем визначений таким незалежним від розташування іменем (з допомогою URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:temporalReferenceSystems:3.2.1

Документ схеми динамічного географічного об'єкту для подання змінних у часі властивостей географічних об'єктів визначений таким незалежним від розташування іменем (з допомогою URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:dynamicFeature:3.2.1

Всі документи схеми перелічені в додатку В.

#### 14.1.2 Зв'язок з ISO 19108

Компоненти часової геометрії та топології GML схеми, визначені в цьому розділі, забезпечують узгоджену часткову реалізацію часової схеми стандарту ISO 19108. Цей зв'язок докладно розглянуто в Г.2.5.

Типи геометрії та топології з ISO 19108, реалізовані у GML, визначені в ISO 19108; деякі додаткові обмеження визначені для цих типів в ISO 19108, що є також обмеженнями на компоненти часової геометрії та топології GML схеми.

Крім цього, GML визначає додаткові компоненти часових схем, описані в Г.3.12 та Г.3.13.

### 14.2 Часова схема

#### 14.2.1 Абстрактні часові об'єкти

##### 14.2.1.1 AbstractTimeObject

gml:AbstractTimeObject реалізує TM\_Object стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, пункт 5.2.2) та діє як голова групи заміни для всіх часових примітивів та комплексів. Він задекларований таким чином:

```
<element name="AbstractTimeObject" type="gml:AbstractTimeObjectType" abstract="true"
substitutionGroup="gml:AbstractGML"/>
```

gml:AbstractTimeObject можна використовувати в будь-якій позиції, де є дійсним gml:AbstractGML. Його модель контенту визначена так:

```
<complexType name="AbstractTimeObjectType" abstract="true">
<complexContent>
<extension base="gml:AbstractGMLType"/>
</complexContent>
</complexType>
```

**Примітка 1.** Модель контенту для gml:AbstractTimeObject є порожнім розширенням типу AbstractGMLType. Типи, отримані з цього, мають стандартні властивості GML об'єкту як наявні: abstractMetadataProperty, description, descriptionReference, name.



#### 14.2.1.2 AbstractTimePrimitive

`gml:AbstractTimePrimitive` впроваджує `TM_Primitive` стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, пункт 5.2.3) та діє як голова групи заміни для геометричних та топологічних часових примітивів. Він задекларований таким чином:

```
<element name="AbstractTimePrimitive" type="gml:AbstractTimePrimitiveType" abstract="true"
substitutionGroup="gml:AbstractTimeObject"/>
```

`gml:AbstractTimePrimitive` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeObject`. Його модель контенту визначена таким чином:

```
<complexType name="AbstractTimePrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTimeObjectType">
      <sequence>
        <element name="relatedTime" type="gml:RelatedTimeType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Це розширює модель для загальних часових об'єктів із властивостями, що показують відношення між цим часовим примітивом та іншими часовими примітивами. Визначення типу `gml:RelatedTimeType` подано в підпункті 14.2.1.4.

#### 14.2.1.3 TimePrimitivePropertyType, validTime

`gml:TimePrimitivePropertyType` забезпечує стандартну модель контенту для асоціацій між довільним членом групи заміни, де головою є `gml:AbstractTimePrimitive`, та іншим об'єктом:

```
<complexType name="TimePrimitivePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractTimePrimitive"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

`gml:validTime` є елементом властивості зручності, задекларований таким чином:

```
<element name="validTime" type="gml:TimePrimitivePropertyType"/>
```

#### 14.2.1.4 RelatedTimeType

`gml:RelatedTimeType` забезпечує модель контенту для показу відносної позиції довільного члена групи заміни, де головою є `gml:AbstractTimePrimitive`. Він розширює загальний тип `gml:TimePrimitivePropertyType` XML атрибутом `relativePosition`, значення якого вибирають з множини 13 часових відношень, ідентифікованих в [Allen,1983]:

```
<complexType name="RelatedTimeType">
  <complexContent>
    <extension base="gml:TimePrimitivePropertyType">
      <attribute name="relativePosition">
        <simpleType>
          <restriction base="string">
            <enumeration value="Before"/>
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>
```

```

    <enumeration value="After"/>
    <enumeration value="Begins"/>
    <enumeration value="Ends"/>
    <enumeration value="During"/>
    <enumeration value="Equals"/>
    <enumeration value="Contains"/>
    <enumeration value="Overlaps"/>
    <enumeration value="Meets"/>
    <enumeration value="OverlappedBy"/>
    <enumeration value="MetBy"/>
    <enumeration value="BegunBy"/>
    <enumeration value="EndedBy"/>
  </restriction>
</simpleType>
</attribute>
</extension>
</complexContent>
</complexType>

```

#### 14.2.1.5 AbstractTimeComplex

`gml:AbstractTimeComplex` є агрегацією часових примітивів та реалізує `TM_Complex` стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, пункт 5.2.2) і діє як голова групи заміни для часових комплексів. Він задекларований таким чином:

```

<element name="AbstractTimeComplex" type="gml:AbstractTimeComplexType" abstract="true"
substitutionGroup="gml:AbstractTimeObject"/>

```

`gml:AbstractTimeComplex` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeObject`.

Його модель контенту визначена таким чином:

```

<complexType name="AbstractTimeComplexType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTimeObjectType"/>
  </complexContent>
</complexType>

```

**Примітка 1.** Цей Міжнародний стандарт тільки визначає часовий комплекс топології. Часовий геометричний комплекс не визначений.

**Примітка 2.** Цей міжнародний стандарт не розрізняє між часовим лінійним графом та часовим нелінійним графом.

### 14.2.2 Часова геометрія

#### 14.2.2.1 Вступ

Часова геометрія описана у термінах моментів часу, періодів, позицій та тривалостей.

#### 14.2.2.2 AbstractTimeGeometricPrimitive

`gml:TimeGeometricPrimitive` реалізує `TM_GeometricPrimitive` стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, пункт 5.2.3) та діє як голова групи заміни для геометричних часових примітивів. Він задекларований таким чином:

```

<element name="AbstractTimeGeometricPrimitive" type="gml:AbstractTimeGeometricPrimitiveType" abstract="true"

```

```
substitutionGroup="gml:AbstractTimePrimitive"/>
```

`gml:AbstractTimeGeometricPrimitive` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimePrimitive`. Його модель контенту визначена таким чином:

```
<complexType name="AbstractTimeGeometricPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTimePrimitiveType">
      <attribute name="frame" type="anyURI" default="#ISO-8601"/>
    </extension>
  </complexContent>
</complexType>
```

Часову геометрію треба асоціювати з часовою референсною системою через атрибут `frame`, що забезпечує URI посилання, яке ідентифікує опис референцної системи. Згідно до стандарту ISO 19108, григоріанський календар з UTC є референсною системою за умовчанням, але інші системи також можна використовувати.

Двома геометричними примітивами в часовому вимірі є момент та період. GML компоненти визначені для підтримки цього (див. наступний підпункт).

#### 14.2.2.3 TimeInstant

`gml:TimeInstant` реалізує `TM_Instant` стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, підпункт 5.2.3.2) та діє як нульвимірний геометричний примітив, що являє позицію в часі, яку можна ідентифікувати. Він задекларований таким чином:

```
<element name="TimeInstant" type="gml:TimeInstantType" substitutionGroup="gml:AbstractTimeGeometricPrimitive"/>
```

`gml:TimeInstant` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeGeometricPrimitive`. Його модель контенту визначена таким чином:

```
<complexType name="TimeInstantType" final="#all">
  <complexContent>
    <extension base="gml:AbstractTimeGeometricPrimitiveType">
      <sequence>
        <element ref="gml:timePosition"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*Приклад 1.* У документі екземпляру `gml:TimeInstant` містить `gml:timePosition` таким чином:

```
<gml:TimeInstant gml:id="t11">
  <gml:description>Abby's birthday</gml:description>
  <gml:timePosition>2001-05-23</gml:timePosition>
</gml:TimeInstant>
```

#### 14.2.2.4 TimeInstantPropertyType

`gml:TimeInstantPropertyType` є спеціалізацією для `gml:TimePrimitivePropertyType`, що забезпечує асоціювання `gml:TimeInstant` з об'єктом:

```
<complexType name="TimeInstantPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeInstant"/>
  </sequence>
</complexType>
```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

#### 14.2.2.5 TimePeriod

`gml:TimePeriod` реалізує `TM_Period` стандарту ISO 19108 (див. Г.2.5.2 та ISO 19108:2002, підпункт 5.2.3.3) та діє як одновимірний геометричний примітив, яким подають екстент у часі, який можна визначити. Він задекларований таким чином:

```

<element name="TimePeriod" type="gml:TimePeriodType"
substitutionGroup="gml:AbstractTimeGeometricPrimitive"/>

```

`gml:TimePeriod` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeGeometricPrimitive`. Його модель контенту визначена таким чином:

```

<complexType name="TimePeriodType">
  <complexContent>
    <extension base="gml:AbstractTimeGeometricPrimitiveType">
      <sequence>
        <choice>
          <element name="beginPosition" type="gml:TimePositionType"/>
          <element name="begin" type="gml:TimeInstantPropertyType"/>
        </choice>
        <choice>
          <element name="endPosition" type="gml:TimePositionType"/>
          <element name="end" type="gml:TimeInstantPropertyType"/>
        </choice>
        <group ref="gml:timeLength" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Розміщення `gml:TimePeriod` у часі описано часовими позиціями моментів, у яких він починається та закінчується. Тривалість періоду дорівнює часовій віддалі між двома обмежувальними часовими позиціями.

І початок, і кінець можна описати в термінах їх прямої позиції з використанням `gml:TimePositionType` (див. підпункт 14.2.2.7), що є простим типом контенту XML схеми, або через посилання на момент часу, який можна ідентифікувати з використанням `gml:TimeInstantPropertyType` (див. підпункт 14.2.2.4).

*Приклад 1.* В межах `gml:TimePeriod` `gml:TimeInstant` може зустрічатися безпосередньо як значення початку та кінця таким чином:

```

<gml:TimePeriod gml:id="p22">
  <gml:begin>
    <gml:TimeInstant gml:id="t11">
      <gml:timePosition>2001-05-23</gml:timePosition>
    </gml:TimeInstant>
  </gml:begin>
  <gml:end>

```

```

<gml:TimeInstant gml:id="t12">
  <gml:timePosition>2001-06-23</gml:timePosition>
</gml:TimeInstant>
</gml:end>
</gml:TimePeriod>

```

Альтернативно для межі періоду `gml:TimePeriod` можна використовувати умовну GML модель властивості для посилання на момент часу, описаний десь іще, або межу треба показати як пряму позицію.

*Приклад 2.* Наступний мішаний приклад показує і перше, і друге, як і включення необов'язкової властивості тривалості `gml:duration`:

```

<gml:TimePeriod gml:id="p22">
  <gml:begin xlink:href="#t11"/>
  <gml:endPosition>2002-05-23</gml:endPosition>
  <gml:duration>P1Y</gml:duration>
</gml:TimePeriod>

```

#### 14.2.2.6 TimePeriodPropertyType

`gml:TimePeriodPropertyType` є спеціалізацією для `gml:TimePrimitivePropertyType`, що забезпечує асоціювання `gml:TimePeriod` із об'єктом:

```

<complexType name="TimePeriodPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimePeriod"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

#### 14.2.2.7 TimePositionType, timePosition

Метод ідентифікації часової позиції є специфічним для кожної часової референційної системи. `gml:TimePositionType` підтримує опис часової позиції згідно до підтипів, описаних в ISO 19108. Він реалізує `TM_Position` стандарту ISO 19108 (див. Г.2.5.5 та ISO 19108:2002, пункт 5.4.2).

У значеннях, базованих на календарях та годинниках, використовують лексичні формати, базовані на стандарті ISO 8601, як описано в XML схемі, частина 2:2001. Можна використовувати десяткове значення з системами координат, такими як GPS-час або UNIX-час. Можна використовувати URI для забезпечення посилань на деяку еру в порядковій референційній системі<sup>3)</sup>.

Звичайно, із багатьма компонентами, змодельованими як типи даних в комплексі міжнародних стандартів ISO 19100, відповідний GML компонент має простий контент. Однак, модель контенту для `gml:TimePositionType` визначена в кілька кроків (деталі переходу в `TM_Position` стандарту ISO 19108 описані в Г.2.5.5):

```

<complexType name="TimePositionType" final="#all">
  <simpleContent>
    <extension base="gml:TimePositionUnion">
      <attribute name="frame" type="anyURI" default="#ISO-8601"/>

```

<sup>3)</sup> наприклад, геологічна епоха

```

<attribute name="calendarEraName" type="string" />
<attribute name="indeterminatePosition" type="gml:TimeIndeterminateValueType" />
</extension>
</simpleContent>
</complexType>

```

Три XML атрибути мають місце на gml:TimePositionType.

Значення часу має бути асоційоване з часовою референчною системою через атрибут frame, що забезпечує URI посилання, яке ідентифікує опис референцної системи. Згідно до 19108, григоріанський календар із UTC є референчною системою за умовчанням, хоча можна використовувати і інші системи. Компоненти для опису часових референцних систем описані в підрозділ 14.4, але не є обов'язковими для опису референцної системи в цей спосіб, оскільки посилання може бути спрямоване на будь-який ресурс, який можна ідентифікувати з допомогою URI.

Для часових значень з використанням календаря, що містить більше ніж одну еру, (необов'язковий) атрибут calendarEraName забезпечує ім'я календарної ери.

Неточні часові позиції можна виразити з використанням необов'язкового атрибуту indeterminatePosition. Він приймає значення з переліку, визначеного таким чином:

```

<simpleType name="TimeIndeterminateValueType">
  <restriction base="string">
    <enumeration value="after"/>
    <enumeration value="before"/>
    <enumeration value="now"/>
    <enumeration value="unknown"/>
  </restriction>
</simpleType>

```

Ці значення інтерпретують таким чином:

- “unknown” (“невідомо”) означає, що не подано жодного специфічного значення для часової позиції;
- “now” (“тепер”) означає, що подане специфічне значення завжди треба замінювати поточною часовою позицією за наявності доступу до значення;
- “before” (“до”) означає, що актуальна часова позиція невідома, але відомо, що вона має бути до визначеного значення;
- “after” (“після”) означає, що актуальна часова позиція невідома, але відомо, що вона має бути після визначеного значення.

Значення для indeterminatePosition може:

- бути використане або окремо, або
- кваліфікувати специфічне значення для часових позицій<sup>4)</sup>.

Простий тип gml:TimePositionUnion є об'єднанням простих XML типів схеми, які забезпечують екземпляри підтипів для часової позиції, описаної в ISO 19108.

```

<simpleType name="TimePositionUnion">
  <union memberTypes="gml:CalDate time dateTime anyURI decimal"/>
</simpleType>

```

<sup>4)</sup> наприклад, перед 2000-12, після 1019624400

На порядкову еру можна посилатися через URI. Десяткове значення можна використовувати для подання віддалі від початку шкали<sup>5)</sup>. `time` використовують для позиції, що щоденно трапляється (див. ISO 19108:2002, підпункт 5.4.4.2).

Нарешті, форми календаря та годинника, що підтримують подання часу в системах, базованих на роках, місяцях, днях, годинах, хвилинах та секундах, зібрані у нотації, що відповідає ISO 8601, таким чином:

```
<simpleType name="CalDate">
  <union memberTypes="date gYearMonth gYear"/>
</simpleType>
```

**Примітка 1.** Простий тип XML схеми `dateTime` не дозволяє скорочення з правого боку, за виключенням випадків дрібних частин секунд, тому потрібні `date`, `gYear` та `gYearMonth`.

**Примітка 2.** Згідно до ISO 19108, коли використовують негригоріанський календар, базований на роках, місяцях, днях, треба використовувати таке ж саме лексичне подання. Згідно до XML схеми (частина 2), ведучі нулі треба додати, якщо значення року інакше матиме менше ніж чотири знаки.

Елемент `gml:timePosition` задекларований таким чином:

```
<element name="timePosition" type="gml:TimePositionType"/>
```

Цей елемент використовують прямо як властивість `gml:TimeInstant` (див. підпункт 14.2.2.3); його можна також використовувати в прикладних схемах.

*Приклад 1.* Наступні приклади показують, як `gml:timePosition` або інші елементи цього типу можуть виступати в екземплярі даних:

```
-<gml:timePosition>2002-11-25T13:20:20</gml:timePosition>;
-<gml:timePosition indeterminatePosition="after">1994</gml:timePosition>;
-<gml:timePosition indeterminatePosition="now">1994-07-10</gml:timePosition>;
-<gml:timePosition name="http://my.big.org/TRS/GPS">25876321.01</gml:timePosition>;
-<gml:timePosition frame="http://my.big.org/TRS/archaeology">
  http://my.history.org/eras/bronzeAge </gml:timePosition>;
-<gml:timePosition frame="http://my.big.org/TRS/calendars/japanese">
  calendarEraName="Meiji">0025-03</gml:timePosition>
```

#### 14.2.2.8 `timeLength`, `duration`, `timeInterval`, `TimeUnitType`

Тривалість часового періоду описано з використанням групи `gml:timeLength`, що задекларована в схемі таким чином:

```
<group name="timeLength">
  <choice>
    <element ref="gml:duration"/>
    <element ref="gml:timeInterval"/>
  </choice>
</group>
```

Її модель контенту є вибором двох елементів властивості:

```
<element name="duration" type="duration"/>
```

<sup>5)</sup> наприклад, UNIX час, GPS календар

`gml:duration` відповідає синтаксису ISO 8601 для часової тривалості, як реалізовано в типі тривалості XML схеми. Іншою альтернативою є `gml:timeInterval`, яка відповідає стандарту ISO/IEC 11404, що базується на значеннях з плаваючою комою для часової тривалості.

```
<element name="timeInterval" type="gml:TimeIntervalLengthType"/>

<complexType name="TimeIntervalLengthType" final="#all">
  <simpleContent>
    <extension base="decimal">
      <attribute name="unit" type="gml:TimeUnitType" use="required"/>
      <attribute name="radix" type="positiveInteger"/>
      <attribute name="factor" type="integer"/>
    </extension>
  </simpleContent>
</complexType>
```

Синтаксис стандарту ISO/IEC 11404 визначає використання додатного цілого числа `positiveInteger` разом із відповідними значеннями для кореня та коефіцієнту. Роздільну здатність інтервалу часу встановлено з точністю до одного  $\text{radix}^{(-\text{factor})}$  визначеної одиниці часу.

*Приклад 1.* Одиниця="секунда", корінь="10", коефіцієнт="3" визначають роздільну здатність у мілісекунду.

Значення одиниці або вибране з одиниць для часових інтервалів із стандарту ISO 80000-3, або є іншою відповідною одиницею. Кодування визначене для GML у `gml:TimeUnitType`:

```
<simpleType name="TimeUnitType">
  <union>
    <simpleType>
      <restriction base="string">
        <enumeration value="year"/>
        <enumeration value="month"/>
        <enumeration value="day"/>
        <enumeration value="hour"/>
        <enumeration value="minute"/>
        <enumeration value="second"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="string">
        <pattern value="other:\w{2,}"/>
      </restriction>
    </simpleType>
  </union>
</simpleType>
```

Другий компонент цього типу об'єднання забезпечує метод для вказання одиниць часу, відмінних від шести стандартних одиниць, поданих у переліку.



*Приклад 1.* Для подання періоду тривалістю у 5 днів, 14 годин та 30 хвилин, прийнятним є будь-який з наступних прикладів.

```
<duration>P5DT14H30M</duration>
```

```
<timeInterval unit="hour" radix="10" factor="0">134.5</timeInterval>
```

```
<timeInterval unit="other:week" radix="10" factor="0"> 0.800595</timeInterval>
```

### **14.3 Часова схема топології**

#### **14.3.1 Вступ**

Часова топологія описана в термінах часових комплексів, вузлів, ребер та зв'язності між ними. Часова топологія безпосередньо не надає інформацію про часову позицію. Її використовують у випадку опису походження або історії (наприклад, дерево родин, що виражає еволюцію видів, екологічний цикл, походження територій та будівель, історія відокремлення та злиття адміністративних меж). У наступних підрозділах визначено часову топологію як часову характеристику об'єктів відповідно до стандарту ISO 19108.

#### **14.3.2 Об'єкти часової топології**

##### **14.3.2.1 Огляд**

Об'єкт часової топології має бути часовим елементом, що описує порядок географічних об'єктів або їх властивостей як часові характеристики цих об'єктів. Два об'єкти часової топології – це примітив та комплекс.

Час є одновимірним топологічним простором, а примітиви часової топології мають бути часовим вузлом, що відповідають екземпляру, та часовим ребром, що відповідає періоду. Часовий вузол є абстракцією події, що трапилась у певний момент, як початок чи кінець одного чи більше станів. Стан є умовою – характеристикою географічного об'єкту чи набору даних, що триває деякий період. “Статичний об'єкт” у цьому стандарті означає об'єкт, що зберігає узгоджений ідентифікатор протягом усього свого життєвого циклу. Часове ребро є абстракцією стану та асоціюється з часовими вузлами, що являють його початок та кінець. Однак, примітиви часової топології безпосередньо не вказують «коли» та «як довго». Часовий вузол не обов'язково є початком або кінцем часового ребра у випадку опису події, що не асоційована із станами. Такий вузол називають ізольованим вузлом.

Топологічний комплекс є колекцією примітивів топології, що є замкненим при операції повернення межі. Часовий топологічний комплекс має бути зв'язаним ациклічним спрямованим графом, що складений із часових ребер та часових вузлів. Мінімальний часовий топологічний комплекс є часовим ребром з двома часовими вузлами на обох його кінцях.

*Приклад 1.* Життєвий цикл будівлі можна описати як послідовність станів: проект, розробка, побудова, експлуатація, передача та знесення. Кожна стадія може бути подана як часове ребро. Межа кожної стадії описана часовим вузлом, яким подають моменти прийняття рішення, на в які припиняється одна стадія та розпочинається інша. Тому життєвий цикл будівлі описаний як часовий топологічний комплекс, що складається з послідовності часових ребер.

##### **14.3.2.2 AbstractTimeTopologyPrimitive**

Примітиви часової топології мають передбачати інформацію, яка забезпечує впорядкування географічних об'єктів та їх властивостей. Часове поєднання об'єктів можна перевірити на наявність примітивів часової топології як значень їх властивостей. Звичайно, швидкоплинний об'єкт асоціюється із часовим вузлом, а статичний об'єкт – із

часовим ребром. Об'єкт з обома режимами асоціюється з часовим примітивом: надтипом часових вузлів та часових ребер.

`gml:TimeTopologyPrimitive` реалізує `TM_TopologicalPrimitive` стандарту ISO 19108 (див. Г.2.5.6 та ISO 19108:2002, підпункт 5.2.4.2) та діє як голова групи заміни примітивів часової топології. Він визначений у схемі таким чином:

```
<element name="AbstractTimeTopologyPrimitive" type="gml:AbstractTimeTopologyPrimitiveType" abstract="true"
  substitutionGroup="gml:AbstractTimePrimitive"/>
```

`gml:AbstractTimeTopologyPrimitive` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimePrimitive`. Модель його контенту визначена так:

```
<complexType name="AbstractTimeTopologyPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTimePrimitiveType">
      <sequence>
<element name="complex" type="gml:ReferenceType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Топологічний примітив завжди зв'язаний із одним чи більше топологічних примітивів, і тому завжди є членом комплексу топології. В GML екземплярі це часто вказано примітивами, що описані елементами, які є нащадками елемента, що описує комплекс. Однак, для підтримки випадків, коли примітив часової топології описаний в іншому контексті, подають необов'язкову властивість `gml:complex`, яка містить посилання на батьківський комплекс часової топології.

#### 14.3.2.3 TimeTopologyPrimitivePropertyType

`gml:TimeTopologyPrimitivePropertyType` забезпечує асоціювання `gml:AbstractTimeTopologyPrimitive` з об'єктом:

```
<complexType name="TimeTopologyPrimitivePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractTimeTopologyPrimitive"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

#### 14.3.2.4 TimeTopologyComplex

Часовий топологічний комплекс має бути зв'язним ациклічним спрямованим графом, що складається з примітивів часової топології, тобто часових вузлів та часових ребер. Оскільки часове ребро не може існувати без двох вузлів на його кінцях, статичні географічні об'єкти мають часові ребра з часового топологічного комплексу як значення їх часових властивостей, незалежно від декларацій у явній формі.

Часовий топологічний комплекс виражає лінійний чи нелінійний граф. Часовим лінійним графом, що складається з послідовності часових ребер, подають розвиток у часі, описаний тільки через «заміну» екземплярів об'єктів або значень елементів

об'єктів. Часовий вузол як початок або кінець графу пов'язаний щонайменш із одним часовим ребром. Часовий вузол, що не є початком та кінцем, має бути пов'язаний щонайменш із двома часовими ребрами, одне з яких починається з вузла, а інше закінчується у вузлі.

`gml:TimeTopologyComplex` реалізує `TM_TopologicalComplex` ISO 19108 (див. Г.2.5.6 та ISO 19108:2002, підпункт 5.2.4.5) та задекларований таким чином:

```
<element name="TimeTopologyComplex" type="gml:TimeTopologyComplexType"
substitutionGroup="gml:AbstractTimeComplex"/>
```

`gml:TimeTopologyComplex` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeComplex`. Його модель контенту визначена таким чином:

```
<complexType name="TimeTopologyComplexType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTimeComplexType">
      <sequence>
<element name="primitive" type="gml:TimeTopologyPrimitivePropertyType" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

Часовий топологічний комплекс є множиною пов'язаних примітивів часової топології. Примітиви, що є його членами, задекларовані або посиланням, або значенням із використанням властивості `gml:primitive`.

#### 14.3.2.5 TimeTopologyComplexPropertyType

`gml:TimeTopologyComplexPropertyType` забезпечує асоціювання `gml:TimeTopologyComplex` з об'єктом:

```
<complexType name="TimeTopologyComplexPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeTopologyComplex"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

#### 14.3.2.6 TimeNode

Часовий вузол є нульвимірним топологічним примітивом, яким подають вузол, що можна ідентифікувати в часі (це є еквівалентом точки в просторі). Вузол може діяти як кінець чи початок будь-якого числа часових ребер. Часовий вузол можна реалізувати як геометрію, його позицію, значенням якої є момент часу.

`gml:TimeNode` реалізує `TM_Node` стандарту ISO 19108 (див. Г.2.5.6 та ISO 19108:2002, підпункт 5.2.4.3) та задекларований таким чином:

```
<element name="TimeNode" type="gml:TimeNodeType"
substitutionGroup="gml:AbstractTimeTopologyPrimitive"/>
```

`gml:TimeNode` можна використовувати в будь-якій позиції, де є дійсним `gml:AbstractTimeTopologyPrimitive`. Його модель контенту визначена таким чином:

```

<complexType name="TimeNodeType">
  <complexContent>
    <extension base="gml:AbstractTimeTopologyPrimitiveType">
      <sequence>
        <element name="previousEdge" type="gml:TimeEdgePropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="nextEdge" type="gml:TimeEdgePropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="position" type="gml:TimeInstantPropertyType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

#### 14.3.2.7 TimeNodePropertyType

gml:TimeNodePropertyType забезпечує асоціювання gml:TimeNode з об'єктом:

```

<complexType name="TimeNodePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeNode"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

#### 14.3.2.8 TimeEdge

Часове ребро є одновимірним топологічним примітивом. Воно є відкритим інтервалом, що починається та закінчується у вузлі. Ребро можна реалізувати як геометрію, значенням якої є часовий період.

gml:TimeEdge реалізує часове ребро TM\_Edge стандарту ISO 19108 (див.Г.2.5.6 та ISO 19108:2002, підпункт 5.2.4.4) та задекларовано таким чином:

```

<element name="TimeEdge" type="gml:TimeEdgeType"
substitutionGroup="gml:AbstractTimeTopologyPrimitive"/>

```

gml:TimeEdge можна використовувати в будь-якій позиції, де є дійсним gml:AbstractTimeTopologyPrimitive. Його модель контенту визначена таким чином:

```

<complexType name="TimeEdgeType">
  <complexContent>
    <extension base="gml:AbstractTimeTopologyPrimitiveType">
      <sequence>
        <element name="start" type="gml:TimeNodePropertyType"/>
        <element name="end" type="gml:TimeNodePropertyType"/>
        <element name="extent" type="gml:TimePeriodPropertyType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

#### 14.3.2.9 TimeEdgePropertyType

gml:TimeEdgePropertyType забезпечує асоціювання gml:TimeEdge з об'єктом:

```

<complexType name="TimeEdgePropertyType">

```

```

<sequence minOccurs="0">
  <element ref="gml:TimeEdge"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

## 14.4 Часові референцні системи

### 14.4.1 Огляд

Значення в часовому домені вимірюють відносно часової референцної системи. Звичайні типи референцних систем включають календарі, порядкові часові референцні системи та часові координатні системи (час вимірюють з початку певної епохи). Первинною часовою референцною системою для використання з географічною інформацією є григоріанський календар та 24-годинний місцевий або узгоджений всесвітній час (Coordinated Universal Time, UTC), але спеціальні аплікації можуть передбачати використання альтернативних референцних систем. Юліанська система нумерації днів є часовою координатною системою, що має початок раніше, ніж будь-який інший відомий календар, опівдні 1 січня 4713 до Р. Х. за юліанським календарем, та є корисною у переході від дат за одним календарем до дат за іншими календарями.

У GML використовують сім конкретних елементів для опису часових референцних систем:

`gml:TimeReferenceSystem`, `gml:TimeCoordinateSystem`, `gml:TimeCalendar`, `gml:TimeCalendarEra`, `gml:TimeClock`, `gml:TimeOrdinalReferenceSystem`, та `gml:TimeOrdinalEra`.

### 14.4.2 Базові часові референцні системи, `TimeReferenceSystem`

Референцну систему характеризують у термінах її домену, в межах якого вона дійсна: просторовий чи часовий екстент, у якому її можна застосовувати. Базовим GML елементом для часової референцної системи є `gml:TimeReferenceSystem`. Його модель контенту розширює `gml:DefinitionType` (див. пункт 15.2.1) однією додатковою властивістю, `gml:domainOfValidity`. Вона реалізована таким чином:

```

<element name="TimeReferenceSystem" type="gml:TimeReferenceSystemType"
  substitutionGroup="gml:Definition"/>

```

`gml:TimeReferenceSystem` можна використовувати в будь-якій позиції, де є дійсним `gml:Definition`. Її модель контенту визначена таким чином:

```

<complexType name="TimeReferenceSystemType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element name="domainOfValidity" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

*Приклад 1.* Цей елемент може зустрічатися в документі екземпляру у такому вигляді:

```

<gml:TimeReferenceSystem gml:id="JulianCalendar">

```

```
<gml:description xlink:href="http://aa.usno.navy.mil/data/docs/JulianDate.html"/>
<gml:name>Julian Calendar</gml:name>
<gml:domainOfValidity>Western Europe</gml:domainOfValidity>
</gml:TimeReferenceSystem>
```

#### 14.4.3 TimeCoordinateSystem

Часова система координат має бути базована на безперервній інтервальній шкалі, визначеної в термінах єдиного часового інтервалу.

gml:TimeCoordinateSystem реалізує TM\_CoordinateSystem стандарту ISO 19108 (див. Г.2.5.9 та ISO 19108:2002, пункт 5.3.3) з очікуваннями, визначеними далі, та задекларована таким чином:

```
<element name="TimeCoordinateSystem" type="gml:TimeCoordinateSystemType"
substitutionGroup="gml:TimeReferenceSystem"/>
```

gml:TimeCoordinateSystem можна використовувати в будь-якій позиції, де є дійсною gml:TimeReferenceSystem. Її модель контенту визначена таким чином:

```
<complexType name="TimeCoordinateSystemType">
  <complexContent>
    <extension base="gml:TimeReferenceSystemType">
      <sequence>
        <choice>
          <element name="originPosition" type="gml:TimePositionType"/>
          <element name="origin" type="gml:TimeInstantPropertyType"/>
        </choice>
        <element name="interval" type="gml:TimeIntervalLengthType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Відмінності від TM\_CoordinateSystem (ISO 19108) включають:

- початок визначений з використанням або властивості gml:originPosition, значення якої є прямою часовою позицією (див. підпункт 14.2.2.7), або властивості gml:origin, моделлю якої є gml:TimeInstantPropertyType (див. підпункт 14.2.2.4); це надає більшій гнучкості у поданні та підтримує посилання на значення, фіксоване десь іще;
- в інтервалі використовують gml:TimeIntervalLengthType, визначений у підпункті 14.2.2.8.

*Приклад 1.* Координатні системи можна описати в екземплярах даних таким чином:

```
<gml:TimeCoordinateSystem gml:id="Laser36">
  <gml:description>Часова шкала, використана під час лазерного експерименту</gml:description>
  <gml:name>Часова шкала лазеру 36</gml:name>
  <gml:domainOfValidity>Лазерна лабораторія</gml:domainOfValidity>
  <gml:origin>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-28T12:50:00+08:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:origin>
  <gml:interval unit="second" radix="10" factor="12">1.0</gml:interval>
```

```
</gml:TimeCoordinateSystem>
```

```
<gml:TimeCoordinateSystem gml:id="geologyMa">
  <gml:name>Геологічна часова система</gml:name>
  <gml:domainOfValidity>Земля</gml:domainOfValidity>
  <gml:origin>
    <gml:TimeInstant>
      <gml:description xlink:href="http://www.c14dating.com/agecalc.html">Умовний початок,
використаний для датування карбону.
Еквівалентний "теперішньому" для інших методів радіометричного датування з набагато
меншою точністю.</gml:description>
      <gml:timePosition>1950</gml:timePosition>
    </gml:TimeInstant>
  </gml:origin>
  <gml:interval unit="year" radix="10" factor="-6">1.0</gml:interval>
</gml:TimeCoordinateSystem>
```

#### 14.4.4 Календарі та годинники

##### 14.4.4.1 Огляд

Календарі та годинники базовані на інтервальних шкалах. Календар є дискретною часовою референчною системою з роздільною здатністю в один день. Годинник забезпечує основу для визначення часової позиції в межах дня. Годинник треба використовувати разом із календарем для забезпечення повного опису часової позиції в межах певного дня.

Календарі мають різні складні внутрішні структури. Ця схема визначає простий зовнішній інтерфейс календаря. Кожен календар надає набір правил для визначення календарної дати з допомогою набору елементів, таких як рік, місяць та день. У будь-якому календарі роки понумеровані відносно дати референчної події, що визначає календарну еру. Один і той же календар може посилатися більше ніж на одну календарну еру.

##### 14.4.4.2 TimeCalendar, TimeCalendarEra

Календар є дискретною часовою референчною системою координат, що надає основу для визначення часової позиції з роздільною здатністю в один день. gml:TimeCalendar реалізує TM\_Calender стандарту ISO 19108 (див. Г.2.5.8 та ISO 19108:2002, підпункт 5.3.2.3) та задекларований таким чином:

```
<element name="TimeCalendar" type="gml:TimeCalendarType" substitutionGroup="gml:TimeReferenceSystem"/>
```

gml:TimeCalendar можна використовувати в будь-якій позиції, де є дійсним gml:TimeReferenceSystem. Його модель контенту визначена таким чином:

```
<complexType name="TimeCalendarType">
  <complexContent>
    <extension base="gml:TimeReferenceSystemType">
      <sequence>
        <element name="referenceFrame" type="gml:TimeCalendarEraPropertyType" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
```

</complexType>

`gml:TimeCalendar` додає одну властивість до тих, що успадковані від `gml:TimeReferenceSystem`. `gml:referenceFrame` забезпечує посилання на `gml:TimeCalendarEra`, яка в ньому використана. `gml:TimeCalendar` може посилатися на більше ніж одну календарну еру.

Елемент `referenceFrame` відповідає стандартній GML моделі властивості, що дозволяє створення екземплярів асоціації або з використанням включеного опису, використовуючи елемент `gml:TimeCalendarEra`, або з використанням посилання на `gml:TimeCalendarEra`, що є наявним десь іще.

`gml:TimeCalendarEra` реалізує `TM_CalenderEra` стандарту ISO 19108 (див. Г.2.5.8 та ISO 19108:2002, підпункт 5.3.2.1) та задекларовано таким чином:

```
<element name="TimeCalendarEra" type="gml:TimeCalendarEraType" />
```

Її модель контенту визначена таким чином:

```
<complexType name="TimeCalendarEraType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element name="referenceEvent" type="gml:StringOrRefType"/>
        <element name="referenceDate" type="gml:CalDate"/>
        <element name="julianReference" type="decimal"/>
        <element name="epochOfUse" type="gml:TimePeriodPropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:TimeCalendarEra` успадковує базові властивості від `gml:DefinitionType` (див. пункт 15.2.1) та має такі додаткові властивості:

- `gml:referenceEvent`, що є іменем або описом міфічної чи історичної події, яка фіксує позицію базової шкали календарної ери; її подають як текст або з використанням посилань на опис, наявний десь іще;
- `gml:referenceDate`, що уточнює дату події `referenceEvent`, виражену як дата у даному календарі; у більшості календарів ця дата є початком (тобто, першим днем) шкали, але не завжди;
- `gml:julianReference` уточнює юліанську дату, що відповідає референційній даті; номер юліанського дня є цілочисловим значенням; юліанська дата є десятковим значенням, що дозволяє кращу роздільну здатність; перерахунок календарних дат в юліанські дати і навпаки надає відносно простий базис для перерахунку дат з одного календаря в інший.
- `gml:epochOfUse` є періодом, для якого використана календарна ера як базис для датування.

#### **14.4.4.3 TimeCalendarPropertyType, TimeCalendarEraPropertyType**

`gml:TimeCalendarPropertyType` забезпечує асоціювання `gml:TimeCalendar` з об'єктом:



```

<complexType name="TimeCalendarPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeCalendar"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

`gml:TimeCalendarEraPropertyType` забезпечує асоціювання `gml:TimeCalendarEra` з об'єктом:

```

<complexType name="TimeCalendarEraPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeCalendarEra"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

#### 14.4.4.4 TimeClock

Годинник забезпечує базис для визначення часової позиції у межах доби. Годинник треба використовувати з календарем для подання повного опису часової позиції в межах певної доби. `gml:TimeClock` реалізує `TM_Clock ISO 19108` (див.Г.2.5.8 та `ISO 19108:2002`, підпункт 5.3.2.4) та задекларований таким чином:

```

<element name="TimeClock" type="gml:TimeClockType"
substitutionGroup="gml:TimeReferenceSystem"/>

```

`gml:TimeClock` можна використовувати в будь-якій позиції, де `gml:TimeReferenceSystem` є дійсним. Його модель контенту визначена таким чином:

```

<complexType name="TimeClockType" final="#all">
  <complexContent>
    <extension base="gml:TimeReferenceSystemType">
      <sequence>
        <element name="referenceEvent" type="gml:StringOrRefType"/>
        <element name="referenceTime" type="time"/>
        <element name="utcReference" type="time"/>
        <element name="dateBasis" type="gml:TimeCalendarPropertyType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:TimeClock` додає такі властивості до успадкованих від `gml:TimeReferenceSystemType`:

- `gml:referenceEvent`, що є іменем або описом події, такої як південь чи схід Сонця, що фіксує позицію базової шкали годинника;
- `gml:referenceTime`, що визначає час доби, асоційований з референсною подією, поданою як час доби за певним годинником; референційний час звичайно є початком шкали годинника;

- gml:utcReference, що уточнює 24-годинний місцевий або всесвітній час, що відповідає референцному часу;
- gml:dateBasis містить календарі, в яких використовують цей годинник, або посилається на них.

#### 14.4.4.5 TimeClockPropertyType

gml:TimeClockPropertyType забезпечує асоціювання gml:TimeClock з об'єктом:

```
<complexType name="TimeClockPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeClock"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

### 14.4.5 Часові порядкові референцні системи

#### 14.4.5.1 Огляд

У деяких галузях застосування географічної інформації – таких, як геологія та археологія – відносна позиція в часі відома точніше ніж абсолютний час чи тривалість. Порядок подій у часі може бути чітко встановлений, але розмір інтервалів між ними неможливо точно визначити; у таких випадках доречним є використання порядкової часової референцної системи. Порядкова часова референцна система складається з послідовності іменованих ер без перерв між ними, що можуть, у свою чергу, складатися з ер нижчого порядку за більш детальної шкали, становлячи в цілому ієрархічну структуру ер різної роздільної здатності.

Порядкова часова референцна система, де ери, що є компонентами, більше не поділяються, є часовим топологічним комплексом, що повинен бути лінійним графом. Порядкова часова референцна система, де деякі або всі ери, що є компонентами, поділяються, є часовим топологічним комплексом з обмеженням, згідно з яким паралельні гілки можуть бути побудовані тільки парами, де одна з яких є суцільною часовою порядковою ерою, а друга – послідовністю часових порядкових ер, які називають членами групи. Це обмеження означає, що в певній часовій порядковій референцній системі відносна позиція порядкових ер є однозначно визначеною.

Позиції початку та кінця даної ери можуть калібрувати відносно часову шкалу.

#### 14.4.5.2 TimeOrdinalReferenceSystem, TimeOrdinalEra

gml:TimeOrdinalReferenceSystem реалізує TM\_OrdinalReferenceSystem стандарту ISO 19108 (див. Г.2.5.10 та ISO 19108:2002, пункт 5.3.4) шляхом додавання однієї чи більше властивостей gml:component до загальної моделі. Її задекларовано таким чином:

```
<element name="TimeOrdinalReferenceSystem" type="gml:TimeOrdinalReferenceSystemType"
  substitutionGroup="gml:TimeReferenceSystem"/>
```

gml:TimeOrdinalReferenceSystem можна використовувати в будь-якій позиції, де є дійсною gml:TimeReferenceSystem. Її модель контенту визначено таким чином:

```
<complexType name="TimeOrdinalReferenceSystemType">
  <complexContent>
    <extension base="gml:TimeReferenceSystemType">
```

```

    <sequence>
      <element name="component" type="gml:TimeOrdinalEraPropertyType" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

gml:TimeOrdinalEra реалізує TM\_OrdinalEra стандарту ISO 19108 (див. Г.2.5.10 та ISO 19108:2002, пункт 5.3.4). Його модель контенту відповідає шаблону gml:TimeEdge (див. підпункт 14.3.2.8), успадковуюючи стандартні властивості від gml:DefinitionType (див. пункт 15.2.1), та додаючи властивості gml:start, gml:end та gml:extent, набір властивостей gml:member, що показують упорядковані елементи gml:TimeOrdinalEra, та властивість gml:group, що вказує на батьківську еру. Його задекларовано таким чином:

```

<element name="TimeOrdinalEra" type="gml:TimeOrdinalEraType"/>

<complexType name="TimeOrdinalEraType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element name="relatedTime" type="gml:RelatedTimeType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="start" type="gml:TimeNodePropertyType"/>
        <element name="end" type="gml:TimeNodePropertyType"/>
        <element name="extent" type="gml:TimePeriodPropertyType" minOccurs="0"/>
        <element name="member" type="gml:TimeOrdinalEraPropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="group" type="gml:ReferenceType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Зворотнє включення елементів gml:TimeOrdinalEra дозволяє побудову ієрархічної порядкової референцної схеми довільної глибини, так, щоб порядкова ера на даному рівні ієрархії включала послідовність більш коротких суміжних порядкових ер.

*Приклад 1.* Нижче показано частину геологічної часової шкали, поданої як порядкова референцна система:

```

<gml:TimeOrdinalReferenceSystem gml:id="GeologicalTimeScale">
  <gml:description xlink:href="ftp://ftp.iugs.org/pub/iugs/iugs_intstratchart.pdf"/>
  <gml:name>Geological time scale</gml:name>
  <gml:domainOfValidity>Earth</gml:domainOfValidity>
  <!-- Earlier eras omitted -->
  <gml:component>
    <gml:TimeOrdinalEra gml:id="Cenozoic">
      <gml:name>Cenozoic Era</gml:name>
      <gml:start xlink:href="#basePaleocene"/>
      <gml:end xlink:href="#now"/>
      <gml:member>
        <gml:TimeOrdinalEra gml:id="Tertiary">
          <gml:name>Tertiary Period</gml:name>

```

```

<gml:start xlink:href="#baseTertiary"/>
<gml:end xlink:href="#basePleistocene"/>
<gml:member>
  <gml:TimeOrdinalEra gml:id="Paleogene">
    <gml:name>Paleogene sub-period</gml:name>
    <gml:start>
      <gml:TimeInstant gml:id="basePaleogene">
        <gml:timePosition frame="#geologyMa">65.0</gml:timePosition>
      </gml:TimeInstant>
    </gml:start>
    <gml:end xlink:href="#baseNeogene"/>
    <gml:member>
      <gml:TimeOrdinalEra gml:id="Paleocene">
        <gml:name>Paleocene Epoch</gml:name>
        <gml:start xlink:href="#basePaleogene"/>
        <gml:end xlink:href="#baseEocene"/>
      </gml:TimeOrdinalEra>
    </gml:member>
    <gml:member>
      <gml:TimeOrdinalEra gml:id="Eocene">
        <gml:name>Paleocene Epoch</gml:name>
        <gml:start >
          <gml:TimeInstant gml:id="baseEocene">
            <gml:timePosition frame="#geologyMa">57.8</gml:timePosition>
          </gml:TimeInstant>
        </gml:start >
        <gml:end xlink:href="#baseOligocene"/>
      </gml:TimeOrdinalEra>
    </gml:member>
    <gml:member>
      <gml:TimeOrdinalEra gml:id="Oligocene">
        <gml:name>Oligocene Epoch</gml:name>
        <gml:start >
          <gml:TimeInstant gml:id="baseOligocene">
            <gml:timePosition frame="#geologyMa">33.7</gml:timePosition>
          </gml:TimeInstant>
        </gml:start >
        <gml:end xlink:href="#baseNeogene"/>
      </gml:TimeOrdinalEra>
    </gml:member>
  </gml:TimeOrdinalEra>
</gml:member>
<!-- Neogene sub-period and Quaternary period omitted -->
</gml:TimeOrdinalEra>
</gml:member>
</gml:TimeOrdinalEra>
</gml:component>
</gml:TimeOrdinalReferenceSystem>

```

Треба зауважити, що використання посилань на різні початкові та кінцеві елементи дозволяють записувати позицію меж між ерами для подальшого багаторазового використання, відповідно до нелінійного графу, якщо потрібно. Всі позиції посилаються на основу “geologyMa”, яку можна було б визначити як часову систему координат (див. розділ 12).

#### 14.4.5.3 TimeOrdinalEraPropertyType

`gml:TimeOrdinalEraPropertyType` забезпечує асоціювання `gml:TimeOrdinalEra` з об’єктом:

```
<complexType name="TimeOrdinalEraPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TimeOrdinalEra"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

### 14.5 Подання динамічних об’єктів

#### 14.5.1 Огляд

Низку типів та відношень визначено для подання змінних у часі властивостей географічних об’єктів.

У загальному розгляді просторово-часового моделювання Лангран (Langran) розрізняв три принципові часові сутності: стани, події та свідок; схема, визначена у наступних підрозділах, включає елементи для кожної з них. Концептуальна модель показана у Г.3.11.

#### 14.5.2 DataSource

У GML, свідка подають простою властивістю `gml:dataSource` чи `gml:dataSourceReference`, яка вказує джерело часових даних.

```
<element name="dataSource" type="gml:StringOrRefType"/>
<element name="dataSourceReference" type="gml:ReferenceType"/>
```

*Атрибут віддалених зв’язків елементу `gml:dataSource` був скасований разом із його поточним типом.*

*Для посилання на джерело віддалених даних натомість треба використовувати атрибути зв’язків `gml:dataSourceReference`.*

*Приклад 1. Людина-спостерігач або сенсор.*

#### 14.5.3 Динамічні властивості

Група зручності `gml:dynamicProperties` визначена таким чином:

```
<group name="dynamicProperties">
  <sequence>
    <element ref="gml:validTime" minOccurs="0"/>
    <element ref="gml:history" minOccurs="0"/>
    <element ref="gml:dataSource" minOccurs="0"/>
    <element ref="gml:dataSourceReference" minOccurs="0"/>
  </sequence>
</group>
```

Це дозволяє включати в прикладну схему динамічні властивості в моделі контенту стандартним шляхом.

Властивість `gml:validTime` визначена в підпункті 14.2.1.3. Інші властивості визначені в інших місцях у підрозділі 14.5.

#### 14.5.4 DynamicFeature

Стани отримують по визначених у часі екземплярах географічного об'єкту. `gml:DynamicFeature` забезпечує реалізацію динамічного географічного об'єкту, як показано у Г.3.1.3 та задекларовано таким чином:

```
<element name="DynamicFeature" type="gml:DynamicFeatureType" substitutionGroup="gml:AbstractFeature"/>
```

Модель контенту розширює стандарт `gml:AbstractFeatureType` групою моделі `gml:dynamicProperties`:

```
<complexType name="DynamicFeatureType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <group ref="gml:dynamicProperties"/>
    </extension>
  </complexContent>
</complexType>
```

Кожний визначений у часі екземпляр представляє знімок географічного об'єкту. Класи динамічного географічного об'єкту звичайно розширюють для того, щоб вони були сумісними з певними аплікаціями. Динамічний географічний об'єкт є носієм або знімку в часі, або історії.

**Примітка 1.** Історія складається з набору часових зрізів `gml:AbstractTimeSlices`, і вони можуть містити будь-які змінні в часі властивості. Можна для прикладу використати такий механізм для опису географічного об'єкту з однією властивістю, яка змінюється в часі.

#### 14.5.5 DynamicFeatureCollection

`gml:DynamicFeatureCollection` реалізує `DynamicFeatureCollection`, як показано в Г.3.13 та задекларовано таким чином:

```
<element name="DynamicFeatureCollection" type="gml:DynamicFeatureCollectionType"
  substitutionGroup="gml:DynamicFeature"/>
```

Модель контенту розширює `gml:DynamicFeatureType` властивістю `gml:dynamicMembers`:

```
<complexType name="DynamicFeatureCollectionType">
  <complexContent>
    <extension base="gml:DynamicFeatureType">
      <sequence>
<element ref="gml:dynamicMembers"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="dynamicMembers" type="gml:DynamicFeatureMemberType"/>
```

```

<complexType name="DynamicFeatureMemberType">
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence>
<element ref="gml:DynamicFeature" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

`gml:DynamicFeatureCollection` є колекцією географічних об'єктів, що має властивість `gml:validTime` (тобто, знімок колекції географічних об'єктів) або яка має властивість `gml:history`, що містить один чи більше історичних зрізів `gml:AbstractTimeSlices`, кожен з яких містить значення змінних у часі властивостей колекції географічних об'єктів. Треба зауважити, що `gml:DynamicFeatureCollection` може бути:

- або географічним об'єктом, що складається із статичних об'єктів-членів (вони не змінюються в часі), але який має властивості об'єкту-колекції як цілого, що змінюються в часі<sup>6)</sup>, наприклад, поїзд, що є колекцією об'єктів: місцезнаходження та швидкість поїзда змінюються в часі та можуть бути подані в історії поїзду; є також статичними властивості вагонів, такі як позиція вагону в поїзді, вантаж, матеріал та тип вагону (зерновий вагон, цистерна і т.і.);
- колекцією географічних об'єктів, що складається з динамічних географічних об'єктів-членів (`gml:DynamicFeatures`), але також має властивості колекції як цілого, що змінюються в часі, наприклад, колекція суден у змаганні яхт (яхти можуть сходити з дистанції та знову повертатись, а акваторія, де відбуваються змагання, може змінюватися в часі).

**Примітка 1.** Бувають також колекції географічних об'єктів з динамічними членами, але властивості колекції в цілому є статичними. Це можна також застосувати до змагання яхт, де є тільки властивості, такі як організаційний комітет, та розташування старту та лінії фінішу.

#### 14.5.6 AbstractTimeSlice

Для опису події, тобто дії, що відбувається миттєво або протягом інтервалу часу, у GML надано елемент `gml:AbstractTimeSlice`, задекларований таким чином:

```

<element name="AbstractTimeSlice" type="gml:AbstractTimeSliceType" abstract="true"
substitutionGroup="gml:AbstractGML"/>

```

```

<complexType name="AbstractTimeSliceType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:validTime"/>
        <element ref="gml:dataSource" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

<sup>6)</sup> наприклад, описаний історією

```
</complexContent>
</complexType>
```

Часовий зріз інкапсулює змінні в часі властивості динамічного географічного об'єкту (його треба розширити для подання фіксованої в часі проекції специфічного географічного об'єкту). Властивість `gml:dataSource` описує, як були отримані часові дані.

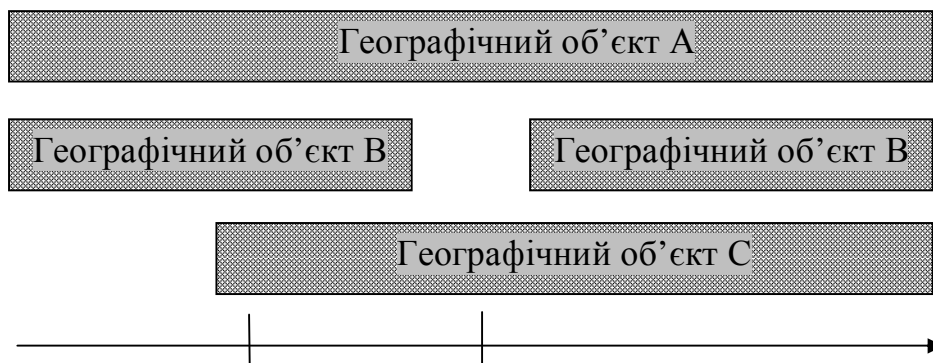
Екземпляр `gml:AbstractTimeSlice` є GML об'єктом, що інкапсулює оновлення динамічних або плинних властивостей, які відображають саме подію зміни, таким чином включає тільки ті властивості географічного об'єкту, що дійсно змінилися внаслідок якогось процесу.

*Приклад 1.* Припустимо, що власність на будинок змінюється, та він перейменований; якщо жодна інша його властивість не змінилася, то ця подія включає оновлене ім'я.

`gml:AbstractTimeSlice` є основою для забезпечення можливостей для фіксації в часі на рівні атрибутів, в протилежність до фіксації в часі екземплярів динамічних географічних об'єктів на рівні об'єктів.

Часовий зріз можна таким чином розглядати як орієнтований на процес чи подію, тоді як знімок є скоріше орієнтованим на стан чи структуру. Часовий зріз має більш багате наповнення, тоді як знімок відображає статус цілого.

*Приклад 2.* Колекція об'єктів може мати життєвий цикл, поданий послідовністю знімків, див. рис. 3.



**Рисунок 3** – Життєвий цикл колекції об'єктів

У момент  $t_1$ , об'єкт А, об'єкт В та об'єкт С є членами колекції. Однак, у момент  $t_2$  тільки об'єкт А та об'єкт С є членами. Докладніше вивчення історії об'єкту В покаже його ефемерність (наприклад, будівлю розбирають та збирають на сезонній основі).

#### 14.5.7 history

Загальна послідовність подій становить `gml:history` об'єкту. Цей елемент властивості задекларовано таким чином:

```
<element name="history" type="gml:HistoryPropertyType"/>
<complexType name="HistoryPropertyType">
  <sequence>
    <element ref="gml:AbstractTimeSlice" maxOccurs="unbounded"/>
  </sequence>
```



```
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Елемент `gml:history` містить набір елементів у групі заміни, очолюваної абстрактним елементом `gml:AbstractTimeSlice`, яким подають властивості, що змінюються в часі. Властивість історії динамічного об'єкту асоціює екземпляр об'єкту з послідовністю часових зрізів (тобто, зміною подій), що інкапсулює еволюцію об'єкту.

**Примітка 1.** Властивість `gml:history` призначена для отримання змінних у часі властивостей об'єкту, ідентичність якого є інваріантною протягом існування часової моделі. У такий спосіб детальну еволюцію об'єкту можна описати так, як його переміщення в просторі або зміни в часі. Властивість `gml:history` може мати відношення до часових об'єктів топології, описаних у підрозділі 14.3. Кожен `gml:AbstractTimeSlice` у `gml:history` динамічного об'єкту може відповідати `gml:TimeEdge` у моделі часової топології, якщо топологія дійсних часів по різних часових зрізах має бути явно подана. У часовій топології будують часовий топологічний комплекс, що забезпечує основу, до якої можна приєднати походження географічного об'єкту або часової колекції географічних об'єктів, включно з динамічними об'єктами.

`gml:MovingObjectStatus` (див. К.7.2) є одним із прикладів, як можна розширити `gml:AbstractTimeSlice` для статусу рухомого об'єкту в певні часи. Цей тип скасовано задля гармонізації з майбутнім ISO 19141 (Схема для рухомих об'єктів).

Якщо географічним об'єктом подають рухомий об'єкт, наприклад, наземний транспортний засіб чи судно, то властивість включає послідовність елементів `gml:MovingObjectStatus`. Динамічний об'єкт, наприклад, циклон, може мати властивість `gml:history`, як показано у наступному фрагменті:

```
<app:Cyclone gml:id="c1">
  <gml:history>
    <gml:MovingObjectStatus>
      <gml:validTime><gml:TimeInstant>
        <gml:timePosition>2005-11-28T13:00:00</gml:timePosition>
      </gml:TimeInstant></gml:validTime>
      <gml:location><gml:Point>
        <gml:pos>140. -35.</gml:pos>
      </gml:Point></gml:location>
      <gml:speed uom="#kph">12.</gml:speed>
      <gml:bearing>
        <gml:CompassPoint>SE</gml:CompassPoint>
      </gml:bearing>
    </gml:MovingObjectStatus>
    <gml:MovingObjectStatus>
      <gml:validTime><gml:TimeInstant>
        <gml:timePosition>2005-11-28T14:00:00</gml:timePosition>
      </gml:TimeInstant></gml:validTime>
      <gml:location><gml:Point>
        <gml:pos>140.1 -34.9</gml:pos>
      </gml:Point></gml:location>
      <gml:speed uom="#kph">23.</gml:speed>
      <gml:bearing>
        <gml:CompassPoint>ESE</gml:CompassPoint>
      </gml:bearing>
  </gml:history>
```

```

</gml:MovingObjectStatus>
</gml:history>
<app:Cyclone>

```

## 15 GML СХЕМА – ВИЗНАЧЕННЯ ТА СЛОВНИКИ

### 15.1 Огляд

Багато аплікацій вимагають визначень термінів, що використовують в документах екземплярів як значення певних властивостей або як референцну інформацію для зв'язування деяким чином властивостей із стандартною інформацією. Одиниці вимірювання та описи явищ, які можна вимірювати, є двома окремими прикладами.

Часто зручно використовувати визначення, подані іншими організаціями. Вони можуть уже бути в пакетах для передачі в різні способи, у тому числі он-лайн та оф-лайн. З тим, щоб на них можна було посилатися з GML документів, взагалі потрібно, щоб URI був наявним у кожному визначенні. У такому випадку звичайно краще посилатися безпосередньо на них.

Альтернативно, може бути зручним або необхідним давати визначення в XML, або вбудовані в документі екземпляру, що містить географічні об'єкти, або як окремий документ. Ці визначення можуть бути транскрипціями з зовнішнього джерела або можуть бути новими визначеннями для місцевого використання. Для реалізації цього випадку подано деякі прості компоненти у GML у таких формах:

- загальний gml:Definition, що може виступати як основа для більш спеціалізованих визначень;
- загальний gml:Dictionary, що підтримує набір визначень або посилань на визначення, які треба зібрати.

Ці компоненти можна використовувати безпосередньо, і вони також виступають як основа для елементів більш спеціалізованих визначень у GML, зокрема, в координатних операціях (розділ 12), координатних референцних системах (розділ 12), датах (розділ 12), часових референцних системах (розділ 14), та одиницях вимірювання (розділ 16).

Треба зауважити, що GML компоненти визначень та словників забезпечують реалізацію простої вкладеної ієрархії визначень з ідентифікаторами. Останнє забезпечує інструменти, які можна використовувати в описі більш складних відношень між термінами. Однак, GML компоненти словника не призначені для забезпечення прямої підтримки складних таксономій, онтологій та тезаурусів. Спеціалізовані XML інструменти надані для задоволення більш складних вимог.

**Примітка 1.** Документ схеми словника ідентифікують таким назалежним від розміщення іменем (з допомогою URNсинтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:dictionary:3.2.1
```

### 15.2 Схема словника

#### 15.2.1 Definition, DefinitionType, remarks

Базовий елемент gml:Definition уточнює визначення, яке може включати або на яке може посилатися словник. Він задекларований таким чином:

```

<element name="Definition" type="gml:DefinitionType"
substitutionGroup="gml:AbstractGML"/>

```

```

<complexType name="DefinitionBaseType">
  <complexContent>
    <restriction base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:descriptionReference" minOccurs="0"/>
        <element ref="gml:identifier"/>
        <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="gml:id" use="required"/>
    </restriction>
  </complexContent>
</complexType>

```

```

<complexType name="DefinitionType">
  <complexContent>
    <extension base="gml:DefinitionBaseType">
      <sequence>
        <element ref="gml:remarks" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```
<element name="remarks" type="string"/>
```

Модель контенту для загального визначення є похідною від `gml:AbstractGMLType`.

Елемент властивості `gml:description` повинен зберігати визначення, якщо його можна помістити у простому текстовому рядку, або елемент властивості `gml:descriptionReference` може забезпечувати посилання на опис десь іще.

Елемент `gml:identifier` має забезпечувати один ідентифікатор для ідентифікації цього визначення. Ідентифікатор має бути унікальним у словниках, де використане це визначення.

Елементи `gml:name` мають забезпечувати нуль чи більше термінів та синонімів, для яких існує це визначення.

Елемент `gml:remarks` треба використовувати для зберігання додаткової текстової інформації, що концептуально не є частиною визначення, але корисна для розуміння визначення.

### 15.2.2 Dictionary, DictionaryType

Набори визначень можна зібрати в словники або колекції. Вони задекларовані в схемі таким чином:

```
<element name="Dictionary" type="gml:DictionaryType" substitutionGroup="gml:Definition"/>
```

```
<complexType name="DictionaryType">
```

```

<complexContent>
  <extension base="gml:DefinitionType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="gml:dictionaryEntry"/>
      <element ref="gml:indirectEntry"/>
    </choice>
    <attributeGroup ref="gml:AggregationAttributeGroup"/>
  </extension>
</complexContent>
</complexType>

```

`gml:Dictionary` є неабстрактною колекцією визначень.

Модель контенту для `gml:Dictionary` додає список властивостей `gml:dictionaryEntry` та `gml:indirectEntry` (скасовані), що містить об'єкти `gml:Definition` або посилається на них. **handle** бази даних (атрибут `gml:id`) треба подавати для того, щоб на колекцію можна було посилатися. Стандартні властивості `gml:identifier`, `gml:description`, `gml:descriptionReference` та `gml:name` наявні для того, щоб містити більше інформації про цей словник або посилатися на неї. Елементи властивості `gml:description` та `gml:descriptionReference` можна використовувати для опису цього словника. Похідний елемент `gml:name` можна використовувати для імені (імен) цього словника.

### 15.2.3 dictionaryEntry, definitionMember, DictionaryEntryType

Ці елементи містять визначення, що є членами словника, або посилаються на них. Елемент `gml:dictionaryEntry` задекларований таким чином:

```

<element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
<complexType name="DictionaryEntryType">
  <complexContent>
    <extension base="gml:AbstractMemberType">
      <sequence minOccurs="0">
        <element ref="gml:Definition"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

Модель контенту відповідає стандартному шаблону GML властивості, тому `gml:dictionaryEntry` може або містити одне `gml:Definition`, або посилатися на нього. Оскільки можна замінити `gml:Definition` на `gml:Dictionary`, контент запису може бути словником нижчого рівня.

Треба зауважити, що, якщо значення надано шляхом посилання, це визначення не несе **handle** (`gml:id`) у цьому контексті, тому не дозволяє зовнішні посилання на це специфічне визначення в цьому контексті. Коли визначення, на яке спрямоване посилання, використовують у цей спосіб, то воно звичайно є в словнику в тому ж самому XML документі.

### 15.2.4 Використання визначень та словників

Словники та визначення є GML об'єктами, тому зустрічаються в незалежних GML документах екземплярів даних.

У прикладних схемах може бути корисним приєднувати gml:Dictionary або gml:Definitions до колекції географічних об'єктів для запису визначень, використаних у властивостях членів колекції.

*Приклад 1.* Два екземпляри словника:

```

<gml:Dictionary gml:id="rockTypes">
  <gml:description>A simple dictionary of rock types using components from gmlBase</gml:description>
  <gml:identifier codeSpace="http://www.abc.org/terms">Rock Types</gml:identifier>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="granite">
      <gml:description>A igneous rock normally composed of quartz, two feldspars and optional mica</gml:description>
      <gml:identifier codeSpace="http://www.abc.org/terms">Granite</gml:identifier>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="sst">
      <gml:description>A detrital sedimentary rock normally composed of siliceous grains</gml:description>
      <gml:identifier codeSpace="http://www.abc.org/terms">Sandstone</gml:identifier>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry xlink:href="http://my.big.org/definitions/geology/limestone"/>
</gml:Dictionary>

<gml:Dictionary gml:id="AbridgedGMLdictionary">
  <gml:description>Abridged GML dictionary.</gml:description>
  <gml:identifier codeSpace="http://www.opengis.net/gml/3.2">GML Dictionary</gml:name>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="term4.1">
      <gml:description>conceptual schema for data required by one or more applications</gml:description>
      <gml:identifier codeSpace="http://www.isotc211.org/19101">application schema</gml:identifier>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="term4.2">
      <gml:description>application schema written in XML Schema according to the rules specified in ISO
19136</gml:description>
      <gml:identifier codeSpace="http://www.opengis.net/gml">GML application schema</gml:identifier>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="term4.3">
      <gml:description>semantic relationship between two or more classifiers that specifies
connections among their instances </gml:description>
      <gml:identifier codeSpace="http://www.uml.org/1.3">association</gml:identifier>
    </gml:Definition>
  </gml:dictionaryEntry>

```

```

<gml:dictionaryEntry>
  <gml:Definition gml:id="term4.4">
    <gml:description>name-value pair contained in an element</gml:description>
    <gml:identifier
codeSpace="http://www.w3.org/XML/1998/namespace">attribute</gml:identifier>
  </gml:Definition>
</gml:dictionaryEntry>
<!-- ... -->
</gml:Dictionary>

```

## 16 GML СХЕМА – ОДИНИЦІ, МІРИ ТА ЗНАЧЕННЯ

### 16.1 Вступ

Кілька компонентів GML схеми стосуються або вимагають кількісних значень, у яких використані референсні шкали або одиниці вимірювання. В підрозділі 8.2 типи `gml:MeasureType`, `gml:MeasureListType` та `gml:MeasureOrNilReasonListType` визначені для уможливлення GML властивостей та об'єктів для того, щоб подавати одиниці вимірювання згідно до такого шаблону:

```
<abc:length uom = "m">100</abc:length>
```

Атрибут `uom` означає “одиниця вимірювання” та має `gml:UomIdentifier` (див. підпункт 8.2.3.6).

У цьому розділі описані компоненти схеми по таких напрямках:

- набір компонентів для опису одиниць вимірювання;
- набір типізованих мір;
- структури для агрегатів та списків мір.

### 16.2 Схема одиниць

#### 16.2.1 Огляд

Деякі компоненти GML схеми стосуються референсних шкал або одиниць вимірювання чи вимагають їх. Одиниці потрібні для кількісних мір, що можуть виступати як значення властивостей типів географічних об'єктів, отриманих по результатах спостережень, у параметрах діапазону покриття, та для мір, використаних у визначеннях референсних систем координат.

**Примітка 1.** Документ схеми `units.xsd` визначає компоненти для підтримки визначення одиниць вимірювання. Схему цих одиниць включено в список в додатку В; вона ідентифікована таким незалежним від розташування іменем (з допомогою URN синтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:units:3.2.1
```

Визначення базової одиниці є розширенням загального елемента `gml:Definition`, визначеного в пункті 15.2.1. Від нього походять три спеціалізовані елементи для визначення одиниці.

Ця модель базована на системі одиниць SI [ISO 1000], в якій розрізняють між базовими одиницями та похідними одиницями:

- базові одиниці – це переважні одиниці для набору ортогональних фундаментальних величин, що визначають спеціалізовані системи одиниць, які не можна отримати через комбінацію інших базових одиниць;

– похідні одиниці – це переважні одиниці для інших величин у системі, які можна визначити через алгебраїчну комбінацію базових одиниць.

У деяких напрямках застосування використовують умовні одиниці, які можна конвертувати у переважні одиниці з використанням коефіцієнту масштабу або формули, що визначає зміну масштабу та зміщення. Набір переважних одиниць для всіх типів фізичних величин у певній системі одиниць складається з об'єднання його базових та похідних одиниць.

### 16.2.2 Використання визначень одиниць

Визначення одиниць може замінювати елемент `gml:Definition`, задекларований як частина моделі словника. Словник, що містить тільки визначення одиниць та посилання на визначення одиниць, є словником одиниць.

### 16.2.3 `unitOfMeasure`, `UnitOfMeasureType`

Елемент `gml:unitOfMeasure` є елементом властивості для посилання на одиницю вимірювання. Він задекларований у схемі таким чином:

```
<element name="unitOfMeasure" type="gml:UnitOfMeasureType"/>

<complexType name="UnitOfMeasureType">
  <sequence/>
  <attribute name="uom" type="gml:UomIdentifier" use="required"/>
</complexType>
```

Це є порожнім елементом, що забезпечує посилання на визначення одиниці вимірювання (див. підпункт 8.2.3.6).

*Приклад 1.* Цей елемент зустрічається в екземплярі даних у такому вигляді:

```
<unitOfMeasure uom="m"/>

<unitOfMeasure uom="http://my.standards.org/units/length/metre"/>
```

### 16.2.4 `UnitDefinition`, `UnitDefinitionType`

`gml:UnitDefinition` є загальним визначенням одиниці вимірювання. Цей загальний елемент використовують тільки для одиниць, для яких немає відомих відношень з іншими одиницями чи системами одиниць. Він задекларований у схемі таким чином:

```
<element name="UnitDefinition" type="gml:UnitDefinitionType"
substitutionGroup="gml:Definition"/>

<complexType name="UnitDefinitionType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element ref="gml:quantityType" minOccurs="0"/>
        <element ref="gml:quantityTypeReference" minOccurs="0"/>
        <element ref="gml:catalogSymbol" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Модель контенту для `gml:UnitDefinition` додає три додаткові властивості до `gml:Definition` (описані в пункті 15.2.1), `gml:quantityType`, `gml:quantityTypeReference` та `gml:catalogSymbol`.

Властивість `gml:catalogSymbol` необов'язково надає короткий символ, який використовують для цієї одиниці. Цей елемент звичайно використовують, коли відношення цієї одиниці до інших одиниць або систем одиниць невідоме.

### 16.2.5 quantityType, quantityTypeReference

Властивості `gml:quantityType` та `gml:quantityTypeReference` вказують на явище, до якого застосовані одиниці. Вони задекларовані таким чином:

```
<element name="quantityType" type="gml:StringOrRefType"/>
```

```
<element name="quantityTypeReference" type="gml:ReferenceType"/>
```

Цей елемент містить неформальний опис явища чи типу фізичної величини, яку вимірюють чи спостерігають.

*Приклади.* "Довжина", "кут", "час", "тиск" чи "температура".

Коли фізична величина є результатом спостереження чи вимірювання, цей термін відомий як тип, який спостерігають, або об'єкт вимірювання.

Використання `gml:quantityType` для посилань на віддалені значення скасоване, натомість треба використовувати `gml:quantityTypeReference`.

### 16.2.6 catalogSymbol

`catalogSymbol` є переважним лексичним символом, який використовують для цієї одиниці вимірювання. Він задекларований таким чином:

```
<element name="catalogSymbol" type="gml:CodeType"/>
```

Атрибут `codeSpace` у `gml:CodeType` ідентифікує простір імен для значення символу каталогу та може посилатися на зовнішній каталог. Значення `string` у `gml:CodeType` містить значення символу, який має бути унікальним у просторі імен каталогу. Цей символ часто зустрічається явним чином у каталозі, але він міг би бути комбінацією символів із допомогою визначеної алгебри одиниць.

*Приклад 1.* Символ "cm" може означати, що це є символ "m", комбінований із префіксом "c".

### 16.2.7 BaseUnit, BaseUnitType, unitsSystem

Базову одиницю вимірювання не можна отримати шляхом комбінації інших базових одиниць у певній системі одиниць. Наприклад, у системі одиниць SI базові одиниці – метр, кілограм, секунда, ампер, кельвін, моль, кандела, а типи фізичних величин – довжина, маса, інтервал часу, електричний струм, термодинамічна температура, обсяг речовини та інтенсивність світла, відповідно.

Це підтримують із допомогою елементу `gml:BaseUnit`, який задекларований таким чином:

```
<element name="BaseUnit" type="gml:BaseUnitType" substitutionGroup="gml:UnitDefinition"/>
```

```
<complexType name="BaseUnitType">
```

```
<complexContent>
```

```
<extension base="gml:UnitDefinitionType">
```

```
<sequence>
```



```

    <element name="unitsSystem" type="gml:ReferenceType"/>
  </sequence>
</extension>
</complexContent>
</complexType>

```

`gml:BaseUnit` розширює загальний `gml:UnitDefinition` властивістю `gml:unitsSystem`, яка забезпечує посилання на систему одиниць, до якої, як доведено, належить ця базова одиниця.

### 16.2.8 DerivedUnit, DerivedUnitType

Похідні одиниці визначені шляхом комбінації інших одиниць. Похідні одиниці використовують для величин, відмінних від тих, що відповідають базовим одиницям, такі як герц ( $\text{с}^{-1}$ ) для частоти, ньютон ( $\text{кг}\times\text{м}/\text{с}^2$ ) для сили. Похідним одиницям, що базуються безпосередньо на базових одиницях, звичайно надають перевагу для величин, відмінних від фундаментальних величин у системі. Якщо похідна одиниця не є переважною одиницею, треба натомість використовувати елемент `gml:ConventionalUnit` (див. пункт 16.2.10). Елемент `gml:DerivedUnit` задекларований таким чином:

```

<element name="DerivedUnit" type="gml:DerivedUnitType"
substitutionGroup="gml:UnitDefinition"/>

<complexType name="DerivedUnitType">
  <complexContent>
    <extension base="gml:UnitDefinitionType">
      <sequence>
        <element ref="gml:derivationUnitTerm" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

`gml:DerivedUnit` розширює `gml:UnitDefinition` властивістю `gml:derivationUnitTerms`.

### 16.2.9 derivationUnitTerms, DerivationUnitTermType

Набір елементів `gml:derivationUnitTerm` описує похідну одиницю вимірювання. Кожен елемент несе цілочисловий показник. Терми скомбіновані шляхом піднесення кожної одиниці, на яку є посилання, до ступеня та формування добутку. Елемент `gml:derivationUnitTerm` задекларований таким чином:

```

<element name="derivationUnitTerm" type="gml:DerivationUnitTermType"/>

<complexType name="DerivationUnitTermType">
  <complexContent>
    <extension base="gml:UnitOfMeasureType">
      <attribute name="exponent" type="integer"/>
    </extension>
  </complexContent>
</complexType>

```

Цей терм одиниці посилається на іншу одиницю вимірювання (uom) та забезпечує цілочисловий показник, застосований до тієї одиниці у визначенні складної одиниці. Показник може бути додатним чи від'ємним, але не нулем.

#### 16.2.10 ConventionalUnit, ConventionalUnitType

Умовні одиниці, що не є ані базовими одиницями, ані визначеними прямою комбінацією базових одиниць, використовують в багатьох апікаційних доменах. Прикладами є електронвольт для енергії, фут та морська миля для довжини. У більшості випадків є відомі, звичайно лінійні, перетворення у переважну одиницю, що є або базова одиниця, або отриманою шляхом прямої комбінації базових одиниць. Елемент `gml:ConventionalUnit` задекларований таким чином:

```
<element name="ConventionalUnit" type="gml:ConventionalUnitType"
substitutionGroup="gml:UnitDefinition"/>

<complexType name="ConventionalUnitType">
  <complexContent>
    <extension base="gml:UnitDefinitionType">
      <sequence>
        <choice>
          <element ref="gml:conversionToPreferredUnit"/>
          <element ref="gml:roughConversionToPreferredUnit"/>
        </choice>
        <element ref="gml:derivationUnitTerm" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

`gml:ConventionalUnit` розширює `gml:UnitDefinition` властивістю, що описує перетворення у переважну одиницю для цієї фізичної величини. Коли перетворення точне, треба використовувати елемент `gml:conversionToPreferredUnit`, або коли перетворення неточне, то наявний елемент `gml:roughConversionToPreferredUnit`. Обидва ці елементи мають одну і ту ж модель контенту. Властивість `gml:derivationUnitTerm`, визначена вище, включена для того, щоб надати користувачеві можливість записувати (необов'язково), як ця одиниця може походити з інших ("більш примітивних") одиниць.

#### 16.2.11 conversionToPreferredUnit, roughConversionToPreferredUnit, ConversionToPreferredUnitType, FormulaType

Елементи `gml:conversionToPreferredUnit` та `gml:roughConversionToPreferredUnit` являють параметри, використані для перетворення умовних одиниць у переважні одиниці для цього типу фізичної величини. Переважна одиниця – це або базова, або похідна одиниця, вибрана для всіх значень одного типу фізичної величини.

Ці перетворення задекларовані в схемі таким чином:

```
<element name="conversionToPreferredUnit" type="gml:ConversionToPreferredUnitType"/>

<element name="roughConversionToPreferredUnit" type="gml:ConversionToPreferredUnitType"/>

<complexType name="ConversionToPreferredUnitType">
```

```

<complexContent>
  <extension base="gml:UnitOfMeasureType">
    <choice>
      <element name="factor" type="double"/>
      <element name="formula" type="gml:FormulaType"/>
    </choice>
  </extension>
</complexContent>
</complexType>

```

Успадкований атрибут uom посилається на переважну одиницю, до якої застосоване це перетворення. Перетворення одиниці в переважну одиницю визначено арифметичним перетворенням (коефіцієнт масштабу та (або) зміщення). Модель контенту розширює gml:UnitOfMeasureType, що має обов'язковий атрибут uom, який ідентифікує переважну одиницю для типу фізичної величини, до якої застосоване перетворення. Перетворення визначене:

- вибором gml:factor, який визначає коефіцієнт масштабу чи
- gml:formula, що визначає формулу, за якими значення з використанням умовної одиниці можна перетворити для отримання відповідного значення з використанням переважної одиниці вимірювання. Модель для формули подана таким чином:

```

<complexType name="FormulaType">
  <sequence>
    <element name="a" type="double" minOccurs="0"/>
    <element name="b" type="double"/>
    <element name="c" type="double"/>
    <element name="d" type="double" minOccurs="0"/>
  </sequence>
</complexType>

```

Ця формула визначає параметри простої формули, за якою значення з використанням умовної одиниці вимірювання можна перетворити у відповідне значення з використанням переважної одиниці вимірювання. Елемент формули містить елементи a, b, c and d, у значеннях яких використаний тип double XML схеми. Ці значення використовують у формулі  $y = (a + bx) / (c + dx)$ , де x є значенням із використанням цієї одиниці, а y є відповідним значенням із використанням базової одиниці. Елементи a та d є необов'язковими, і якщо значення не надані, ці параметри вважають рівними нулю. Якщо значення не надані для a, ані для d, то формула зводиться до дробі з параметрами чисельника та знаменника.

#### 16.2.12 Приклади словника одиниць (довідкові)

Цей словник містить визначення відповідно до всіх базових та похідних одиниць, визначених у системі одиниць SI, та вибірку умовних одиниць для ілюстрації використання цих компонентів:

```

<gml:Dictionary gml:id="unitsDictionary">
  <gml:description>Словник одиниць виміру</gml:description>
  <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">OWS-1.2
Units</gml:identifier>
  <gml:dictionaryEntry>

```

```

<gml:Dictionary gml:id="SIBaseUnits">
  <gml:description>Базові одиниці з системи одиниць SI.</gml:description>
  <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">OWS-1.2 SIBase Units</gml:identifier>
  <gml:dictionaryEntry>
    <gml:BaseUnit gml:id="m">
<gml:description>Метр є довжиною шляху, який світ долає у вакуумі протягом інтервалу часу в
1/299792458 секунди.</gml:description>
      <gml:identifier codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">metre</gml:identifier>
      <gml:name xml:lang="en/US">meter</gml:name>
      <gml:quantityType>length</gml:quantityType>
      <gml:catalogSymbol codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">m</gml:catalogSymbol>
      <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
    </gml:BaseUnit>
  </gml:dictionaryEntry>
  <!-- ... -->
</gml:Dictionary>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
  <gml:Dictionary gml:id="SIDerivedUnits">
    <gml:description>Похідні одиниці з системи одиниць SI. Вони всі отримані як добуток
базових одиниць SI, за виключенням градусів Цельсія, в яких у формулі перетворення в базову
одiniцю SI Base Unit (кельвін) враховано зміщення. </gml:description>
    <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">OWS-1.2 SI
Похідні одиниці</gml:identifier>
    <gml:dictionaryEntry><gml:DerivedUnit gml:id="rad">
      <gml:identifier codeSpace="http://www.bipm.fr/en/3_SI">radian</gml:identifier>
      <gml:quantityType>plane angle</gml:quantityType>
      <gml:catalogSymbol codeSpace="http://www.bipm.fr/en/3_SI">rad</gml:catalogSymbol>
      <gml:derivationUnitTerm uom="#m" exponent="1"/>
      <gml:derivationUnitTerm uom="#m" exponent="-1"/>
    </gml:DerivedUnit>
    </gml:dictionaryEntry>
    <!-- ... -->
  </gml:Dictionary>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
  <gml:Dictionary gml:id="ConventionalUnitsDictionary">
<gml:description>Колекція умовних одиниць. Вони є одиницями виміру, що або широко
використовують, або які важливі в специфічній спільноті. Для більшості з них є відоме
походження від більш примітивних одиниць, що можуть бути чи не бути базовими одиницями
S, або відоме перетворення у переважну одиницю, яка може бути чи не бути базовою одиницею
SI або похідною одиницею, шляхом зміни масштабу та (або) зміщення.</gml:description>
    <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">OWS-1.2 Умовні
одiniці.</gml:identifier>
    <gml:dictionaryEntry>
      <gml:DerivedUnit gml:id="m3">
        <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">cubic metre</gml:identifier>
        <gml:quantityType>Volume</gml:quantityType>
        <gml:derivationUnitTerm uom="#m" exponent="3"/>

```

```

</gml:DerivedUnit>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
  <gml:ConventionalUnit gml:id="l">
    <gml:identifier codeSpace="http://www.opengeospatial.org/initiatives/?iid=79">litre</gml:identifier>
    <gml:quantityType>Volume</gml:quantityType>
    <gml:conversionToPreferredUnit uom="#m3">
      <gml:factor>0.001</gml:factor>
    </gml:conversionToPreferredUnit>
  </gml:ConventionalUnit>
</gml:dictionaryEntry>
<!-- ... -->
</gml:Dictionary>
</gml:dictionaryEntry>
</gml:Dictionary>

```

### 16.3 Схема мір

#### 16.3.1 Огляд

`gml:MeasureType` визначений у схемі базових типів. Типи міри, визначені тут, відповідають набору типів зручності для міри, описаному в ISO/TS 19103. XML реалізація, базована на простому типі XML схеми “double”, який підтримує як десяткову, так і наукову нотації та включає XML атрибут “uom”, що посилається на одиниці вимірювання для значення. Треба зауважити, що немає вимоги зберігати значення з використанням специфічного формату, та аплікації, що отримують елементи цього типу, можуть за вибором перекладати дані в будь-який інший тип, якщо це зручно.

**Примітка 1.** Документ схеми для специфічних типів міри визначений незалежним від розміщення іменем (з використанням URN синтаксису):

```
urn: x-ogc:specification:gml:schema-xsd:measures:3.2.1
```

#### 16.3.2 measure

Це є значення фізичної величини, разом із її одиницями. Вона задекларована таким чином:

```
<element name="measure" type="gml:MeasureType"/>
```

Див. підпункт 8.2.3.6 для визначення `gml:MeasureType`.

#### 16.3.3 Скалярні типи міри

Набір специфічних типів міри визначений у порожніх розширеннях (тобто, альтернативних іменах) `gml:MeasureType`. Визначення-прототип має такий вигляд:

```

<complexType name="LengthType">
  <simpleContent>
    <extension base="gml:MeasureType"/>
  </simpleContent>
</complexType>

```

Ця модель контенту підтримує опис величини довжини (чи віддалі) разом із її одиницями. Одиницю міри, на яку посилається `uom`, має підходити для довжини, наприклад, метри чи фути.

Інші типи міри, визначені згідно до цього зразку, включають: `gml:ScaleType`, `gml:GridLengthType`, `gml:AreaType`, `gml:VolumeType`, `gml:SpeedType`, `gml:TimeType`, та `gml:AngleType`.

*Приклад 1.* Елементи, в яких використані ці моделі контенту, можуть зустрічатися в екземплярі даних у такому вигляді:

```
<my:length uom="m">1.76</my:length>
```

```
<my:scale uom="#percent">20.</my:scale>
```

```
<my:gridLength uom="#pixelSpacing">480</my:gridLength>
```

```
<my:gridLength uom="#imageHeight">0.0020833333333333</my:gridLength>
```

```
<my:area uom="#ha">1.76</my:area>
```

```
<my:volume uom="l">0.45</my:volume>
```

```
<my:speed uom="#kmph">73.0</my:speed>
```

```
<gml:angle uom="#gradians">95.</gml:angle>
```

```
<my:time uom="#minutes">30.</my:time>
```

**Примітка 1.** Треба зауважити, що останній елемент у прикладі адресує ті ж самі функціональні вимоги, що й елементи в групі заміни `gml:AbstractTimeLength`, визначеній у розділі 14.

### 16.3.4 angle

Елемент властивості `gml:angle` використовують для запису значення величини кута одним числом разом із його одиницями. Він задекларований таким чином:

```
<element name="angle" type="gml:AngleType"/>
```

## 16.4 Схема об'єктів значення

### 16.4.1 Вступ

Елементи, задекларовані в цьому розділі, побудовані на інших GML компонентах схеми, зокрема `gml:AbstractTimeObject`, `gml:AbstractGeometry`, та таких типах: `gml:MeasureType`, `gml:MeasureListType`, `gml:CodeType`, `gml:CodeOrNilReasonListType`, `gml:BooleanOrNilReasonListType`, `gml:IntegerOrNilReasonList`.

Особливий інтерес становлять елементи, що є головами груп заміни, та одна група, яку називають групою вибору. Вони є головним підставами для схеми об'єктів значення, оскільки вони можуть діяти як змінні у визначенні моделей контенту, такі, як спостереження, коли є бажаним дозволити альтернативні типи значень, деякі з яких можуть мати складний контент, такі, як масиви, об'єкти геометрії та часу, та, де є корисним, наперед не призначати діючий тип значення. Члени груп включають величини, класифікації категорій, булевські рахункові, часові та просторові значення та агрегати, складені з них.

**Примітка 1.** Документ схеми valueObjects.xsd, що описує компоненти для загальних значень, включений у список у додатку В. Він ідентифікований незалежним від розміщення іменем (з використанням URN синтаксису):

urn:x-ogc:specification:gml:schema-xsd:valueObjects:3.2.1

**Примітка 2.** Елементи, задекларовані в цій схемі, використовують для прямого подання значень. Їх моделі контенту взагалі не походять від gml:AbstractGMLType, та вони не забезпечують ідентифікатор.

#### 16.4.2 Ієрархія елементів значень

Об'єкти значення визначені в ієрархії. Концептуальна модель показана в Г.3.17.

Визначені такі відношення:

- конкретні елементи gml:Quantity, gml:Category, gml:Count та gml:Boolean можуть замінити абстрактний елемент gml:AbstractScalarValue;
- конкретні елементи gml:QuantityList, gml:CategoryList, gml:CountList та gml:BooleanList можуть замінити абстрактний елемент gml:AbstractScalarValueList;
- конкретний елемент gml:ValueArray може замінити конкретний елемент gml:CompositeValue;
- абстрактні елементи gml:AbstractScalarValue та gml:AbstractScalarValueList та конкретні елементи gml:CompositeValue, gml:ValueExtent, gml:CategoryExtent, gml:CountExtent та gml:QuantityExtent можуть замінити абстрактний елемент gml:AbstractValue;
- абстрактні елементи gml:AbstractValue, gml:AbstractTimeObject та gml:AbstractGeometry та конкретний елемент gml:Null (*скасований*) є в групі вибору, яку називають gml:Value та використовують для створення композитних відношень у gml:CompositeValue та gml:ValueExtent;
- у схемах, що потребують значення, можна використовувати абстрактний елемент gml:AbstractValue у моделі контенту для того, щоб дозволити появу в екземплярах будь-чого з gml:AbstractScalarValues, gml:AbstractScalarValueLists, gml:CompositeValue чи gml:ValueExtent, або поименованої групи gml:Value для того, щоб дозволити елементи gml:AbstractTimeObject, gml:AbstractGeometry, та gml:Null (скасовано).

#### 16.4.3 Boolean, BooleanList

Призначені для запису значення або списку значень з логіки двох значень із використанням булевського типу XML схеми. Для цих елементів використані такі декларації:

```
<element name="Boolean" substitutionGroup="gml:AbstractScalarValue" nillable="true">
  <complexType>
    <simpleContent>
      <extension base="boolean">
        <attribute name="nilReason" type="gml:nilReasonType"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="BooleanList" type="gml:booleanOrNilReasonList"
substitutionGroup="gml:AbstractScalarValueList"/>
```

gml:booleanOrNilReasonList описаний у підпункті 8.2.4.1.

*Приклад 1.* В екземплярі можуть зустрічатися наступні приклади:

```
<gml:Boolean>1</gml:Boolean>
```

```
<gml:Boolean>>false</gml:Boolean>
```

```
<gml:Boolean xsi:nil="true" nilReason="missing"/>
```

```
<gml:BooleanList>1 missing 0 1 1 http://my.big.org/explanations/theDogAteIt 0 1</gml:BooleanList>
```

**Примітка 1.** На цих прикладах показано використання різних булевських значень {1, 0, true, false}, і той факт, що значення nilReason, такі як “missing” (“відсутнє”) або URI, можна також вставити в список.

#### 16.4.4 Category, CategoryList

Надані для запису термінів, якими подають класифікацію. У цих елементах використані такі декларації схеми:

```
<element name="Category" substitutionGroup="gml:AbstractScalarValue" nillable="true">
  <complexType>
    <simpleContent>
      <extension base="gml:CodeType">
        <attribute name="nilReason" type="gml:NilReasonType"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
```

```
<element name="CategoryList" type="gml:CodeOrNilReasonListType"
substitutionGroup="gml:AbstractScalarValueList"/>
```

gml:Category має необов’язковий XML атрибут codeSpace, значення якого є URI, що ідентифікує словник, список кодів та організацію для цього терміну.

*Приклад 1.* В екземплярі можуть зустрічатися такі приклади:

```
<gml:Category>good</gml:Category>
```

```
<gml:Category xsi:nil="true" nilReason="missing"/>
```

```
<gml:Category codeSpace="http://my.big.org/dictionaries/rocktypes">Сієніт </gml:Category>
```

```
<gml:CategoryList codeSpace="http://my.big.org/dictionaries/rocktypes">Сієніт Граніт відсутній
Туф</gml:CategoryList>
```

```
<gml:CategoryList codeSpace="http://my.big.org/species">лев леопард ягуар</
gml:CategoryList>
```

#### 16.4.5 Count, CountList

Надані для запису цілих чисел, якими подають кількість випадків або темп. У цих елементах використані такі декларації схеми:



```

<element name="Count" substitutionGroup="gml:AbstractScalarValue" nillable="true">
  <complexType>
    <simpleContent>
      <extension base="integer">
        <attribute name="nilReason" type="gml:nilReasonType"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="CountList" type="gml:integerOrNilReasonList"
substitutionGroup="gml:AbstractScalarValueList"/>

```

*Приклад 1.* В екземплярі можуть зустрічатися такі приклади:

```
<gml:Count>513</gml:Count>
```

```
<gml:Count xsi:nil="true" nilReason="missing"/>
```

```
<gml:CountList>34 56 2 inapplicable 153</gml:CountList>
```

#### 16.4.6 Quantity, QuantityList

Надані для запису числових значень із шкалою. Модель контенту елемента є обсягом із використанням типу довге дійсне число XML схеми, що дозволяє десяткову чи наукову нотацію. У цих елементах використані такі декларації схеми:

```

<element name="Quantity" substitutionGroup="gml:AbstractScalarValue" nillable="true">
  <complexType>
    <simpleContent>
      <extension base="gml:MeasureType">
        <attribute name="nilReason" type="gml:nilReasonType"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="QuantityList" type="gml:MeasureOrNilReasonListType"
substitutionGroup="gml:AbstractScalarValueList"/>

```

При цьому потрібен XML атрибут uom (“одиниця вимірювання”), значення якого є URI, який ідентифікує визначення пропорційної шкали одиниць, за якою числові значення будуть помножені, або інтервальної чи позиційної шкали, на якій зустрічається значення.

*Приклад 1.* В екземплярі можуть зустрічатися такі приклади:

```
<gml:Quantity uom="m">4.32e-4</gml:Quantity>
```

```
<gml:Quantity xsi:nil="true" nilReason="withheld"/>
```

```
<gml:QuantityList uom="#C">21. 37. withheld 25.</gml:QuantityList>
```

#### 16.4.7 AbstractValue, AbstractScalarValue, AbstractScalarValueList

`gml:AbstractValue` є абстрактним елементом, що діє як голова групи заміни, яка містить `gml:AbstractScalarValue`, `gml:AbstractScalarValueList`, `gml:CompositeValue` та `gml:ValueExtent`, і (транзитивно) елементи в їх групах заміни.

`gml:AbstractScalarValue` є абстрактним елементом, що діє як голова групи заміни, яка містить `gml:Boolean`, `gml:Category`, `gml:Count` та `gml:Quantity`, і (транзитивно) елементи в їх групах заміни.

`gml:AbstractScalarValueList` є абстрактним елементом, що діє як голова групи заміни, яка містить `gml:BooleanList`, `gml:CategoryList`, `gml:CountList` та `gml:QuantityList`, і (транзитивно) елементи в їх групах заміни.

В цих елементах використані такі декларації схеми:

```
<element name="AbstractValue" type="anyType" abstract="true"
substitutionGroup="gml:AbstractObject"/>
```

```
<element name="AbstractScalarValue" type="anyType" abstract="true"
substitutionGroup="gml:AbstractValue"/>
```

```
<element name="AbstractScalarValueList" type="anyType"
abstract="true"substitutionGroup="gml:AbstractValue"/>
```

Ці елементи можна використовувати в прикладних схемах як змінні, щоб забезпечити появу в XML документі екземпляру будь-якого члена його групи заміни.

#### 16.4.8 Value

Це є групою зручності вибору, що уніфікує загальні значення, визначені в цьому розділі, з просторовим та часовими об'єктами та мірами, описаними вище, щоб будь-що з них можна було використовувати в агрегатних значеннях. У цьому елементі використані такі декларації схеми:

```
<group name="Value">
  <choice>
    <element ref="gml:AbstractValue"/>
    <element ref="gml:AbstractGeometry"/>
    <element ref="gml:AbstractTimeObject"/>
    <element ref="gml:Null"/>
  </choice>
</group>
```

#### 16.4.9 valueProperty, valueComponent, valueComponents

Це є елементи, що забезпечують екземпляри GML властивості, яка містять значення (`Value`) чи посилаються посилається на нього (них). У цьому елементі використані такі декларації схеми:

```
<element name="valueProperty" type="gml:ValuePropertyType"/>
```

```
<element name="valueComponent" type="gml:ValuePropertyType"/>
```

```
<complexType name="ValuePropertyType">
  <sequence minOccurs="0">
    <group ref="gml:Value"/>
  </sequence>
</complexType>
```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="valueComponents" type="gml:ValueArrayPropertyType"/>

<complexType name="ValueArrayPropertyType">
  <sequence maxOccurs="unbounded">
    <group ref="gml:Value" />
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

Слід зауважити, що і `gml:ValuePropertyType`, і `gml:ValueArrayPropertyType` мають групу, яку називають `gml:Value`, в якості їх контенту. Це означає, що будь-який із елементів у групі вибору `gml:Value`, або у групі заміни членів групи вибору може зустрічатись в якості контенту значення властивості.

Елемент `gml:valueProperty` є елементом зручності загального використання. Елементи `gml:valueComponent` та `gml:valueComponents` використовують спеціально в створенні композитних асоціацій.

#### 16.4.10 Composite Value

`gml:CompositeValue` є агрегатним значенням, побудованим із інших значень. Воно містить нуль або довільне число елементів `gml:valueComponent` та нуль або один елемент властивості `gml:valueComponents`. Його можна використовувати для строго парних агрегатів (вектори, тензори) або для довільних колекцій значень. У цьому елементі використані такі декларації схеми:

```

<element name="CompositeValue" type="gml:CompositeValueType"
substitutionGroup="gml:AbstractValue"/>

<complexType name="CompositeValueType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:valueComponent" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:valueComponents" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AggregationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

```

*Приклад 1.* У екземплярі `gml:CompositeValue` можуть зустрічатися такі приклади:

```

<gml:CompositeValue>
  <gml:valueComponent>
    <gml:QuantityList uom="#C">21. 37. withheld 25.</gml:QuantityList>
  </gml:valueComponent>
  <gml:valueComponent>

```

```

    <gml:Category>good</gml:Category>
  </gml:valueComponent>
  <gml:valueComponent>
    <gml:Count xsi:nil="true" nilReason="missing"/>
  </gml:valueComponent>
  <gml:valueComponents>
    <gml:Point srsName="um:x-ogc:def:crs:EPSG:6.6:4326"><gml:pos>71. -32.</gml:pos></gml:Point>
    <gml:Point srsName="um: x-ogc:def:crs:EPSG:6.6:4326"><gml:pos>70. -35.</gml:pos></gml:Point>
    <gml:Point srsName="um: x-ogc:def:crs:EPSG:6.6:4326"><gml:pos>74. -37.</gml:pos></gml:Point>
  </gml:valueComponents>
</gml:Composite Value>

<gml:Composite Value>
  <gml:valueComponents>
    <gml:Point srsName="um:ogc:def:crs:EPSG:6.6:4326"><gml:pos>-67.563 -13.834</gml:pos></gml:Point>
    <gml:Quantity uom="#km">632.</gml:Quantity>
    <gml:TimeInstant><gml:timePosition>1994-06-09T00:33:16.4</gml:timePosition></gml:TimeInstant>
    <gml:Quantity uom="#mom">-1.00</gml:Quantity>
    <gml:Quantity uom="#mom">0.92</gml:Quantity>
    <gml:Quantity uom="#mom">0.09</gml:Quantity>
    <gml:Quantity uom="#mom">-1.69</gml:Quantity>
    <gml:Quantity uom="#mom">-0.09</gml:Quantity>
    <gml:Quantity uom="#mom">-0.37</gml:Quantity>
  </gml:valueComponents>
</gml:Composite Value>

```

#### 16.4.11 ValueArray

Масив значень (Value Array) використовують для однорідних масивів примітивних та агрегатних значень.

Значеннями-членами можуть бути скалярні, композитні значення, масиви та списки. У цьому елементі використані такі декларації схеми:

```

<element name="ValueArray" type="gml:ValueArrayType"
substitutionGroup="gml:CompositeValue"/>

```

```

<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink" xml:lang="en">
  <sch:title>Schematron constraints for GML / ISO 19136</sch:title>
  <sch:ns prefix="sch" uri="http://purl.oclc.org/dsdl/schematron"/>
  <sch:ns prefix="gml" uri="http://www.opengis.net/gml/3.2"/>
  <sch:pattern>
    <sch:rule context="gml:ValueArray">
      <sch:assert test="not(@codeSpace and @uom)">ValueArray may not carry both a reference to a
codeSpace and a uom</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern>
    <sch:rule context="gml:ValueArray">

```

```

=> <sch:assert test="count(gml:valueComponent/*) = count(gml:valueComponent/*[name()
=
name(..../gml:valueComponent[1]/*[1]))">All components shall be of the same type</sch:assert>
=> <sch:assert test="count(gml:valueComponents/*) = count(gml:valueComponents/*[name()
=
name(../*[1]))">All components shall be of the same type</sch:assert>
</sch:rule>
</sch:pattern>
</sch:schema>

```

```

<complexType name="ValueArrayType">
  <complexContent>
    <extension base="gml:CompositeValueType">
      <attributeGroup ref="gml:referenceSystem"/>
    </extension>
  </complexContent>
</complexType>

<attributeGroup name="referenceSystem">
  <attribute name="codeSpace" type="anyURI" />
  <attribute name="uom" type="gml:UomIdentifier" />
</attributeGroup>

```

Масив значень (ValueArray) має ту ж саму модель контенту, що і композитне значення (CompositeValue), але значення-члени мають бути однорідними. Декларація елемента містить обмеження Schematron, яке точно виражає це обмеження. Оскільки члени однорідні, gml:referenceSystem (uom, codeSpace) може бути визначено на gml:ValueArray саму по собі та успадковано всіма членами, якщо потрібно.

*Приклад 1.* Елемент gml:ValueArray може зустрічатись у екземплярах наступним чином. У першому прикладі набір точок, кожна з яких є значенням властивості gml:valueComponent. Одне з цих значень забезпечене шляхом посилання, з використанням стандартного синтаксису xlink:href:

```

<gml:ValueArray>
  <gml:valueComponent>
    <gml:Point srsName="urn:x-ogc:def:crs:EPSG:6.6:63266405">
      <gml:pos>-32. 71.</gml:pos>
    </gml:Point>
  </gml:valueComponent>
  <gml:valueComponent>
    <gml:Point srsName="urn:x-ogc:def:crs:EPSG:6.6:63266405">
      <gml:pos>-35. 70.</gml:pos>
    </gml:Point>
  </gml:valueComponent>
  <gml:valueComponent xlink:href="http://my.big.org/locations/points/point456"/>
</gml:ValueArray>

```

*Приклад 2.* У другому прикладі набір величин міститься у властивості gml:valueComponents property. Одне із значень не є наявним та указане нульовим значенням:

```

<gml:ValueArray>
  <gml:valueComponents>
    <gml:Quantity uom="#C">21.</gml:Quantity>
    <gml:Quantity uom="#C">37.</gml:Quantity>
    <gml:Quantity xsi:nil="true" nilReason="missing"/>
  </gml:valueComponents>
</gml:ValueArray>

```

*Приклад 3.* Треба зауважити, що `gml:AbstractScalarValueList` є переважним для масивів або скалярних значень, оскільки це кодування є більш ефективним. Інформацію в попередньому прикладі можна виразити таким чином:

```

<gml:QuantityList uom="#C">21. 37. missing</gml:QuantityList>

```

Однак, якщо значення компонентів не є скалярними, то потрібна явна форма.

#### **16.4.12 Typed ValueExtents: CategoryExtent, CountExtent, QuantityExtent**

Ці три елементи забезпечені для екстенсів друкованих значень, для категорій, рахунків та величин. Їх моделі контенту визначені шляхом обмеження відповідних типів скалярних списків для того, щоб містити саме два пункти таким чином:

```

<element name="CategoryExtent" type="gml:CategoryExtentType"
substitutionGroup="gml:AbstractValue"/>

```

```

<complexType name="CategoryExtentType">
  <simpleContent>
    <restriction base="gml:CodeOrNilReasonListType">
      <length value="2"/>
    </restriction>
  </simpleContent>
</complexType>

```

```

<element name="CountExtent" type="gml:CountExtentType"
substitutionGroup="gml:AbstractValue"/>

```

```

<simpleType name="CountExtentType">
  <restriction base="gml:integerOrNilReasonList">
    <length value="2"/>
  </restriction>
</simpleType>

```

```

<element name="QuantityExtent" type="gml:QuantityExtentType"
substitutionGroup="gml:AbstractValue"/>

```

```

<complexType name="QuantityExtentType">
  <simpleContent>
    <restriction base="gml:MeasureOrNilReasonListType">
      <length value="2"/>
    </restriction>
  </simpleContent>
</complexType>

```

Елемент `gml:QuantityExtent` чи інший елемент з використанням цього типу міститиме два значення та шкалу.

*Приклад 1.* `<gml:QuantityExtent uom="#mm">0.9.5</gml:QuantityExtent>`

Елемент типу `gml:CategoryExtentType` корисний, якщо простір кодів (`codeSpace`) визначає набір упорядкованих термінів.

*Приклад 2.* `<my:AgeRange codeSpace="http://iugg.org/geologicalPeriods">Cambrian Devonian</my:AgeRange>`

Будь-який екстент значень може описувати інтервал з одним кінцем через використання значення `NilReason` для однієї з його меж.

*Приклад 3.* `<gml:CountExtent>53 inapplicable</gml:CountExtent>` описує цілі числа, починаючи з 53.

### **16.4.13 BooleanPropertyType, CategoryPropertyType, CountPropertyType, QuantityPropertyType**

Набір типів зручності (`gml:BooleanPropertyType`, `gml:CategoryPropertyType`, `gml:CountPropertyType`, `gml:QuantityPropertyType`) наданий для властивостей, контент яких є специфічним членом групи заміни `gml:AbstractScalarValue`. Їх визначення відповідають тому ж самому шаблону, як це видно на прикладі визначення `gml:BooleanPropertyType`:

```
<complexType name="BooleanPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:Boolean"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

## **17 GML СХЕМА – НАПРЯМКИ**

### **17.1 Схема напрямків**

Компоненти схеми напрямків забезпечують розробника GML схеми елементом стандартної властивості для опису напрямку та асоційованими об'єктами, які можна використовувати для подання орієнтації, напрямку, схилення та інших аспектів напрямку географічних об'єктів.

**Примітка 1.** Документ відповідної схеми ідентифікований таким незалежним від розміщення іменем (з використанням URN синтаксису):

`urn:x-ogc:specification:gml:schema-xsd:direction:3.2.1`

### **17.2 direction, DirectionPropertyType**

Властивість `gml:direction` призначена як наперед визначена властивість, якою подають напрямок для призначення географічним об'єктам, визначеним у прикладній схемі. Вона задекларована таким чином:

```
<element name="direction" type="gml:DirectionPropertyType"/>

<complexType name="DirectionPropertyType">
  <sequence minOccurs="0">
    <choice>
      <element name="DirectionVector" type="gml:DirectionVectorType"/>
      <element name="DirectionDescription" type="gml:DirectionDescriptionType"/>
    </choice>
  </sequence>
</complexType>
```

```

    <element name="CompassPoint" type="gml:CompassPointEnumeration"/>
    <element name="DirectionKeyword" type="gml:CodeType"/>
    <element name="DirectionString" type="gml:StringOrRefType"/>
  </choice>
</sequence>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

Два альтернативних види визначення напрямку, вектор або опис, визначені в наступних підрозділах.

### 17.3 DirectionVectorType

Вектори напрямку визначені шляхом забезпечення компонентів вектору таким чином:

```

<complexType name="DirectionVectorType">
  <choice>
    <element ref="gml:vector"/>
    <sequence>
      <element name="horizontalAngle" type="gml:AngleType"/>
      <element name="verticalAngle" type="gml:AngleType"/>
    </sequence>
  </choice>
</complexType>

```

Елемент gml:vector описаний у підпункті 10.1.4.5.

*Приклад 1.* Ця форма може зустрічатися в екземплярі даних таким чином:

```

<gml:direction>
<gml:DirectionVector>
  <gml:vector srsName="#wgs84">0.0 45.0</gml:vector>
</gml:DirectionVector>
</gml:direction>

```

*Використання альтернативного подання через кути скасоване, натомість треба використовувати gml:vector.*

### 17.4 DirectionDescriptionType

Описи напрямку визначені кодом компасної точки, ключовим словом, текстовим описом або посиланням на опис. Елемент gml:DirectionDescriptionType задекларований таким чином:

```

<complexType name="DirectionDescriptionType">
  <choice>
    <element name="compassPoint" type="gml:CompassPointEnumeration"/>
    <element name="keyword" type="gml:CodeType"/>
    <element name="description" type="string"/>
    <element name="reference" type="gml:ReferenceType"/>
  </choice>
</complexType>

```

gml:compassPoint визначено простим переліком рядкового типу, задекларованим таким чином:



```

<simpleType name="CompassPointEnumeration">
  <restriction base="string">
    <enumeration value="N"/>
    <enumeration value="NNE"/>
    <enumeration value="NE"/>
    <enumeration value="ENE"/>
    <enumeration value="E"/>
    <enumeration value="ESE"/>
    <enumeration value="SE"/>
    <enumeration value="SSE"/>
    <enumeration value="S"/>
    <enumeration value="SSW"/>
    <enumeration value="SW"/>
    <enumeration value="WSW"/>
    <enumeration value="W"/>
    <enumeration value="WNW"/>
    <enumeration value="NW"/>
    <enumeration value="NNW"/>
  </restriction>
</simpleType>

```

Ці напрямки завжди приблизні та мають значення з точністю до 22.5°. Тому взагалі не є необхідним визначати референцну основу, хоча це можна деталізувати у визначенні GML прикладної мови.

*Приклад 1.* Ця форма може зустрічатися в екземплярі даних таким чином:

```

<gml:direction>
  <gml:DirectionDescription>
    <gml:compassPoint>WNW</gml:compassPoint>
  </gml:DirectionDescription>
</gml:direction>

```

У додаток, надано елементи, що мають містити описи напрямку, базованих на тексті.

Якщо напрямок визначений із використанням термінів із списку, треба використовувати `gml:keyword`, та список, у якому вказано значення атрибуту `codeSpace`.

*Приклад 2.* Ця форма може зустрічатися в екземплярі даних таким чином:

```

<gml:direction>
  <gml:DirectionDescription>
    <gml:keyword codeSpace="http://my.big.org/terms/direction">onshore</gml:keyword>
  </gml:DirectionDescription>
</gml:direction>

```

Якщо напрямок описаний прозою, треба використовувати `gml:direction` або `gml:reference`, що дозволяє включати значення або посилатися на нього.

*Приклад 3.* Ця форма може зустрічатися в екземплярі даних таким чином:

```

<gml:direction>
  <gml:DirectionDescription>
    <gml:direction>Towards the lighthouse</gml:direction>
  </gml:DirectionDescription>
</gml:direction>

```

```
</gml:DirectionDescription>
</gml:direction>
```

```
<gml:direction>
  <gml:DirectionDescription>
    <gml:reference xlink:href="http://my.big.org/logbook/20021127/paragraph6"/>
  </gml:DirectionDescription>
</gml:direction>
```

## 18 GML СХЕМА – СПОСТЕРЕЖЕННЯ

### 18.1 Спостереження

GML спостереження є моделлю акту спостереження, часто з камерою, особою чи іншим засобом. Географічний об'єкт спостереження описує метадані, зв'язані з моментом сприйняття інформації, разом із значенням результату спостереження. Воно охоплює широке коло випадків, включно з туристичним фото (власне, акт фотографування, а не саме фото).

**Примітка 1.** Ця схема призначена передусім для того, щоб обслуговувати “прості” спостереження. Схеми для наукових, технічних та інженерних спостережень звичайно потребують розробки GML прикладної схеми для таких спостережень. Дивіться, наприклад, специфікацію спостережень та вимірювань Відкритого геопросторового консорціуму.

### 18.2 Схема спостереження

#### 18.2.1 Огляд

У цьому розділі описані два види спостережень, gml:Observation та gml:DirectedObservation.

**Примітка 1.** Спостереження описані в документі схеми observations.xsd. Схема ідентифікована таким незалежним від розміщення іменем (з використанням URN синтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:observation:3.2.1
```

#### 18.2.2 Observation

Елемент gml:Observation задекларований у схемі таким чином:

```
<element name="Observation" type="gml:ObservationType"
substitutionGroup="gml:AbstractFeature"/>
```

```
<complexType name="ObservationType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:validTime"/>
        <element ref="gml:using" minOccurs="0"/>
        <element ref="gml:target" minOccurs="0"/>
        <element ref="gml:resultOf"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Модель контенту є безпосереднім розширенням від `gml:AbstractFeatureType`; вона автоматично має властивості `gml:identifier`, `gml:description`, `gml:descriptionReference`, `gml:name`, та `gml:boundedBy`.

Елемент `gml:validTime` задекларований у підпункті 14.2.1.3. У цьому контексті він описує час спостереження. Треба зауважити, що це може бути момент або період часу.

*Приклад 1.* Деякі приклади простих спостережень виражені таким чином:

```
<gml:Observation>
  <gml:validTime>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:validTime>
  <gml:using xlink:href="http://www.my.org/sensors/thermometer4"/>
  <gml:target xlink:href="http://www.environment.org/stations/1456"/>
  <gml:resultOf>
    <gml:Quantity uom="#C">18.4</gml:Quantity>
  </gml:resultOf>
</gml:Observation>

<gml:Observation>
  <gml:validTime>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:validTime>
  <gml:using xlink:href="http://www.my.org/cameras/leica2"/>
  <gml:subject xlink:href="http://www.tourist.org/sights/mountain3"/>
  <gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
</gml:Observation>

<gml:Observation>
  <gml:validTime>
    <gml:TimeInstant>
      <gml:timePosition>2002-10-25T11:37:25</gml:timePosition>
    </gml:TimeInstant>
  </gml:validTime>
  <gml:subject xlink:href="http://www.people.org/kids/abby"/>
  <gml:resultOf xlink:href="myDaughtersPortrait.jpg"/>
</gml:Observation>
```

### 18.2.3 using

Властивість `gml:using` містить опис сенсору, інструменту або процедури, яку застосовують для спостереження, або посилається на них. Вона задекларована таким чином:

```
<element name="using" type="gml:ProcedurePropertyType"/>

<complexType name="ProcedurePropertyType">
  <sequence minOccurs="0">
```

```

    <element ref="gml:AbstractFeature"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

#### 18.2.4 target

Властивість `gml:target` містить пробу, район або станцію, що є об'єктом спостереження, або посилається на них. Цей елемент властивості задекларований у схемі таким чином:

```

<element name="target" type="gml:TargetPropertyType"/>

<element name="subject" type="gml:TargetPropertyType" substitutionGroup="gml:target">

<complexType name="TargetPropertyType">
  <choice minOccurs="0">
    <element ref="gml:AbstractFeature"/>
    <element ref="gml:AbstractGeometry"/>
  </choice>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

Ця властивість зокрема корисна для дистанційних спостережень, таких, як фотографування, де загальну властивість місцеположення можна застосувати до місцеположення камери або поля спостереження, і тому може бути неоднозначність.

Елемент `gml:subject` наданий як зручний синонім для `gml:target`. Цей термін є загальноживаним у фотографії.

**Примітка 1.** `gml:Observation` не містить наперед визначеної властивості місцеположення. Якщо розробник схеми бажає визначити конкретне місцеположення точки спостереження (місцеположення приймача), то використає властивість місцеположення, тобто точку із значенням. У випадку, коли ціль має відомий напрямок, але невідому віддаль до точки спостереження (дистанційне зондування), то треба використовувати `gml:DirectedObservation`. Якщо відносний напрямок та віддаль відомі, то треба використовувати `gml:DirectedObservationAtDistance`.

*Приклад 1.* Тип географічного об'єкту спостереження, визначений аплікацією, з місцеположенням точки спостереження, можна визначити так:

```

<element name="ObservationWithSensorLocation" type="app:ObservationWithSensorLocationType"
substitutionGroup="gml:Observation"/>
<complexType name="ObservationWithSensorLocationType">
  <complexContent>
    <extension base="gml:ObservationType">
      <sequence>
        <element name="positionOfSensor" type="gml:PointPropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

### 18.2.5 resultOf

Властивість `gml:resultOf` вказує результат спостереження. Значення може бути включене або посиланням на значення десь іще. Вона задекларована у схемі таким чином:

```
<element name="resultOf" type="gml:ResultType"/>

<complexType name="ResultType">
  <sequence minOccurs="0">
    <any namespace="##any"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

*Приклад 1.* Результуюча властивість разом із `gml:DataBlock` для запису спостереження температури та тиску.

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponents>
        <Temperature uom="um:x-
si:v1999:uom:degreesC">template</Temperature>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:valueComponents>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:tupleList>3,101.2</gml:tupleList>
</gml:DataBlock>
```

### 18.2.6 DirectedObservation

`gml:DirectedObservation` є тим же самим, що і спостереження, за виключенням того, що воно додає додаткову властивість `gml:direction`. Вона є напрямком, в якому здійснювалося спостереження. Природно, що його застосовують тільки до певних типів спостережень, таких як візуальне спостереження оком, або спостереження з допомогою наземних камер.

```
<element name="DirectedObservation" type="gml:DirectedObservationType"
  substitutionGroup="gml:Observation"/>

<complexType name="DirectedObservationType">
  <complexContent>
    <extension base="gml:ObservationType">
      <sequence>
        <element ref="gml:direction"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*Приклад:*

```
<gml:DirectedObservation>
  <gml:validTime>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:validTime>
  <gml:using xlink:href="http://www.my.org/cameras/leica2"/>
  <gml:target xlink:href="http://www.tourist.org/sights/mountain3"/>
  <gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
  <gml:direction>
    <gml:DirectionVector>
      <gml:vector srsName="urn:x-ogc:def:crs:EPSG:6.3:4326">0.0 45.0</gml:vector>
    </gml:DirectionVector>
  </gml:direction>
</gml:DirectedObservation>
```

### 18.2.7 DirectedObservationAtDistance

`gml:DirectedObservationAtDistance` додає додаткову властивість віддаль. Вона є віддаллю між спостерігачем та об'єктом спостереження. Природно, що її застосовують тільки до певних типів спостережень, таких як спостереження оком, або спостереження з допомогою наземних камер.

```
<element name="DirectedObservationAtDistance" type="gml:DirectedObservationAtDistanceType"
  substitutionGroup="gml:DirectedObservation"/>
```

```
<complexType name="DirectedObservationAtDistanceType">
  <complexContent>
    <extension base="gml:DirectedObservationType">
      <sequence>
        <element name="distance" type="gml:MeasureType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*Приклад:*

```
<gml:DirectedObservationAtDistance>
  <gml:validTime>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:validTime>
  <gml:using xlink:href="http://www.my.org/cameras/leica2"/>
  <gml:subject xlink:href="http://www.tourist.org/sights/mountain3"/>
  <gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
  <gml:direction>
    <gml:DirectionVector>
```

```

    <gml:vector srsName="urn:x-ogc:def:crs:EPSG:6.3:4326">0.0 45.0</gml:vector>
  </gml:DirectionVector>
</gml:direction>
  <gml:distance uom="m">16500.</gml:distance>
</gml:DirectedObservationAtDistance>

```

## 19 GML СХЕМА – ПОКРИТТЯ

### 19.1 Модель та подання покриття

#### 19.1.1 Загальні зауваження

У цьому розділі визначено GML кодування для покриттів; воно відповідає концептуальній моделі, описаній в ISO 19123.

В ISO 19123 подано визначення:

*Покриття підтримують перехід від просторово-часового домену до значень атрибутів, де типи атрибутів є спільними для всіх географічних позицій у координатному просторі. Прикладами покриттів є растри, нерегулярні мережі трикутників, точкові покриття та полігональні покриття. Покриття є основними структурами даних у низці напрямків застосування, таких як дистанційне зондування, метеорологія, а також картографування рельєфу, ґрунтів та рослинності.*

Інформація, що описує покриття, може бути умовно подана у два способи:

- як набір пар позиція-значення;
- як опис просторово-часового домену (мультигеометрія, GRID) та як опис множини значень діапазону разом із методом чи правилом (що може бути неявним), що призначає значення з множини діапазону кожній позиції в домені.

Перший метод застосовують до доменів, що поділені на дискретні компоненти. Це подання можна реалізувати в GML як однорідну колекцію географічних об'єктів (тобто, всі об'єкти мають той же набір властивостей), де множина місцеположень від об'єктів складає домен, а множина значень властивості складає діапазон. Перехід від домену до діапазону тривіальний: властивості кожного об'єкту призначаються місцеположенню цього об'єкту. Для покриттів, домен яких складається з великої множини позицій, це явне подання може, однак, виявитися громіздким.

Другий метод є більш гнучким з допомогою кількох шляхів:

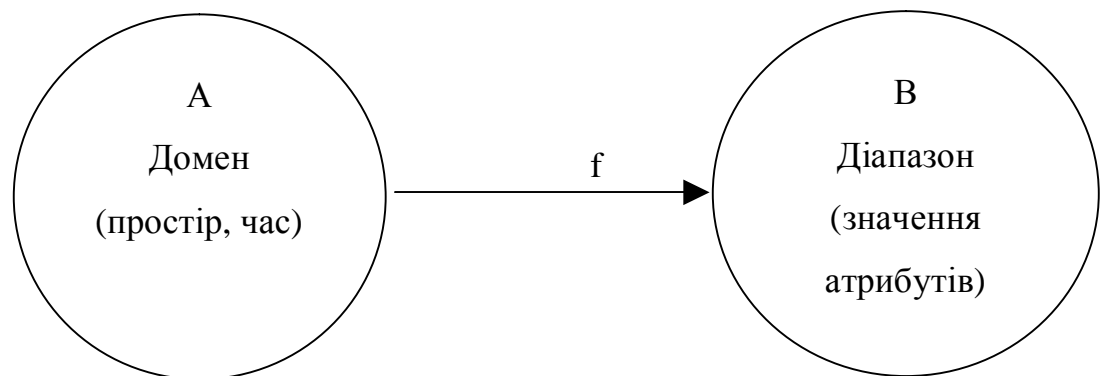
- оскільки домен та діапазон є однорідними множинами, може бути ефективним подання і тільки домену чи діапазону, і обох разом;
- значення в діапазоні можна радше подавати в аналітичній формі, ніж як дискретні явні значення, котрі також зв'язані з фактом як дискретні явні значення;
- коли значення атрибутів безперервно змінюються в межах домену, функціональна форма, що покриває весь домен, потрібна для забезпечення значень діапазону в довільних місцеположеннях (функція звичайно включає інтерполяцію, можливо з використанням моделі процесу).

Перше подання звичайно використовують під час збору даних, де набір або властивості, що стосуються певного місцеположення, використовують разом, або оновлення сховища даних, де тільки невелику кількість географічних об'єктів обробляють одночасно. Друге подання більш зручне для аналізу, де розглядають просторово-часові закономірності та аномалії певної властивості.

Це є другий метод, коли використовують функціональну основу на всьому домені, на якому виконують GML-кодування покриття.

### 19.1.2 Формальний опис покриття

Покриття передбачає перехід від просторово-часового домену до діапазонної множини, яка забезпечує множину, в якій існують значення атрибутів. Діапазонна множина може бути довільною множиною, що включає дискретні списки, цілочислові та десяткові діапазони та багатовимірні векторні простори. Концептуальна модель покриття описана на рисунку 4.



**Рисунок 4** – Концептуальна модель покриття

Покриття можна розглядати як граф функції покриття:  $A \rightarrow B$ , що є множиною впорядкованих пар:

$$\{(x, f(x)) \mid x \text{ входить до } A\}$$

Цей вигляд можна застосувати зокрема до GML кодування покриття. У випадку дискретного покриття множина домену  $A$  поділена на колекцію підмножин (звичайно, розірвана колекція)  $A = \cup A_i$ , та функція  $f$  константна на кожному  $A_i$ . Для просторового домену  $A_i$  є елементами геометрії, тому покриття можна розглядати як колекцію пар (геометрія, значення), де значення є елементом множини діапазону. Якщо просторовий домен  $A$  є топологічним простором, то покриття можна розглядати як колекцію пар (топологія, значення), де елемент топології в парі є топологічним  $n$ -ланцюгом (в термінах GML це є `gml:TopoPoint`, `gml:TopoCurve`, `gml:TopoSurface` або `gml:TopoSolid`).

### 19.1.3 Покриття в GML

Покриття реалізоване як GML географічний об'єкт. Можна говорити про “географічний об'єкт розподілу температур” або про “географічний об'єкт розподілу ґрунтів”, або про “географічний об'єкт, спостережений шляхом дистанційного зондування”.

Як і у випадку будь-якого GML об'єкту, покриття може також бути значенням властивості географічного об'єкту.

*Приклад 1.* Розподіл температури може бути властивістю географічного об'єкту місто (`abc:City`), тому опис міста Оттава можна подати в GML таким чином (тут, `abc:TemperatureCoverage` є об'єктом покриття, що є властивістю об'єкту місто):



```

<abc:City gml:id = "Ottawa">
  <abc:population>500000</abc:population>
  <abc:temperatureDistribution>
    <abc:TemperatureCoverage> ... </abc:TemperatureCoverage>
  </abc:temperatureDistribution>
</abc:City>

```

**Примітка 1.** Покриття в GML підтримують двома документами схеми, coverage.xsd та grids.xsd. Документи coverage.xsd забезпечують базову GML модель покриття. Документ grids.xsd забезпечує структури геометрії, що використані в опису мережевих покриттів, але можуть бути задіяні для інших аплікацій.

Документ схеми grids.xsd ідентифікований таким незалежним від розміщення іменем (із використанням URN синтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:grids:3.2.1
```

Документ схеми coverage.xsd ідентифікований таким незалежним від розміщення іменем (із використанням URN синтаксису):

```
urn:x-ogc:specification:gml:schema-xsd:coverage:3.2.1
```

Всі документи схеми перераховані в додатку В.

#### 19.1.4 Зв'язок з ISO 19123

Компоненти покриття GML схеми, визначені в цьому розділі, забезпечують узгоджене часткове впровадження схеми покриття стандарту ISO 19123. Зв'язок докладно розглянуто в Г.2.11.

Типи покриття стандарту ISO 19123, реалізовані в GML, визначені в ISO 19123; додаткові обмеження, визначені в ISO 19123 для цих типів, є також обмеженнями на компоненти покриття GML схеми.

### 19.2 Схеми мереж

#### 19.2.1 Огляд

Неявний опис геометрії є таким, у якому особливості геометрії не зустрічаються в явному вигляді в кодуванні. Натомість, у компактній нотації записаний набір параметрів, і можна генерувати набір об'єктів із використанням правила із цими параметрами. У цьому пункті надано мережеві геометрії, використані в описі мережевих покриттів та в інших аплікаціях.

У GML визначені дві мережеві структури, а саме gml:Grid та gml:RectifiedGrid.

#### 19.2.2 Grid

gml:Grid реалізує мережеве покриття CV\_Grid стандарту ISO 19123 (див. Г.2.11 та ISO 19123:2005, підрозділ 8.3) та визначено таким чином:

```
<element name="Grid" type="gml:GridType" substitutionGroup="gml:AbstractImplicitGeometry"/>
```

```
<element name="AbstractImplicitGeometry" type="gml:AbstractGeometryType" abstract="true"
substitutionGroup="gml:AbstractGeometry"/>
```

```

<complexType name="GridType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>

```

```

<element name="limits" type="gml:GridLimitsType"/>
<choice>
  <element name="axisLabels" type="gml:NCNameList"/>
  <element name="axisName" type="string" maxOccurs="unbounded"/>
</choice>
</sequence>
<attribute name="dimension" type="positiveInteger" use="required"/>
</extension>
</complexContent>
</complexType>

```

`gml:Grid` неявно визначає нетрансформовану мережу, що є мережею (грідом), складеною двома чи більше наборами ліній, в яких члени кожного набору перетинають членів іншого набору в алгоритмічний спосіб. Область розгляду в межах мережі поданий у термінах меж `gml:limits`, що є GRID-координатами діагонально протилежних кутів прямокутного регіону. `gml:axisLabels` поданий із списком написів осей гріду (`gml:axisName` скасовано). `gml:dimension` визначає розмірність мережі.

У GML елемент `gml:limits` містить єдиний `gml:GridEnvelope`, згідно до таких визначень схеми:

```

<complexType name="GridLimitsType">
  <sequence>
    <element name="GridEnvelope" type="gml:GridEnvelopeType"/>
  </sequence>
</complexType>

<complexType name="GridEnvelopeType">
  <sequence>
    <element name="low" type="gml:integerList"/>
    <element name="high" type="gml:integerList"/>
  </sequence>
</complexType>

```

Кожний з елементів `gml:low` та `gml:high` є `gml:integerLists`, що є кортежами координат; координати вимірюють як зміщення від початку мережі вздовж кожної осі діагонально протилежних кутів прямокутної області розгляду.

*Приклад 1.* Наступні приклади демонструють просту мережу:

```

<gml:Grid dimension="2">
  <gml:limits>
<gml:GridEnvelope>
  <gml:low>0 0</gml:low>
  <gml:high>3 3</gml:high>
</gml:GridEnvelope>
</gml:limits>
  <gml:axisLabels>x y</gml:axisLabels>
</gml:Grid>

```

В цьому прикладі мережа має контрольні точки (пости) в координатах (0,0), (0,1),(1,0),(1,1) до (3,3).

### 19.2.3 RectifiedGrid

Трансформована мережа є мережею, для якої виконано афінну трансформацію між координатами мережі та координатами зовнішньої референційної системи координат. Її визначено шляхом уточнення позиції (у деякому геометричному просторі) початку мережі та векторів, які визначають позиції контрольних точок.

`gml:RectifiedGrid` реалізує трансформовану мережу `CV_RectifiedGrid` стандарту ISO 19123 (див. Г.2.11 та ISO 19123:2005, підрозділ 8.9) та задекларована таким чином:

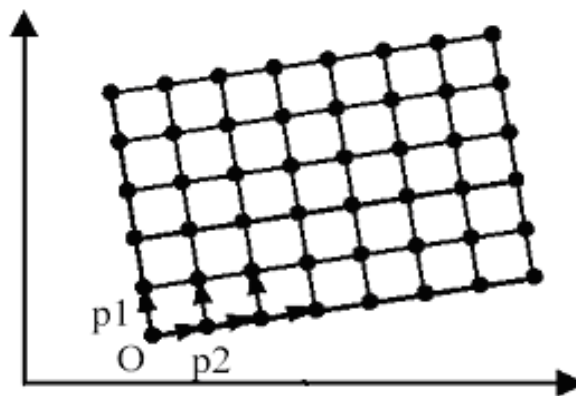
```
<element name="RectifiedGrid" type="gml:RectifiedGridType" substitutionGroup="gml:Grid"/>
```

```
<complexType name="RectifiedGridType">
  <complexContent>
    <extension base="gml:GridType">
      <sequence>
        <element name="origin" type="gml:PointPropertyType"/>
        <element name="offsetVector" type="gml:VectorType" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Слід зауважити, що межі мережі (індекси контрольних точок) та властивості імен осей успадковані від `gml:GridType`, та що `gml:RectifiedGrid` додає властивість `gml:origin` (містить `gml:Point` або посилання на нього) та набір властивостей `gml:offsetVector` (визначених із використанням `gml:VectorType` як його тип даних, як описано в підпункті 10.1.4.5).

**Примітка 1.** `gml:origin` та список властивостей `gml:offsetVector` зв'язують мережу з позицією у географічному просторі та вказують зміщення комірок вздовж кожної осі. Дивіться список обмежень на ці властивості в ISO 19123:2005, пункт 8.9.6.

*Приклад 1.* На рис. 5 показано геометрію трансформованої мережі.



**Рисунок 5** – Геометрія прив'язаної регулярної мережі:

$p_1$ ,  $p_2$  – вектори зміщення,  $O$  – початок мережі

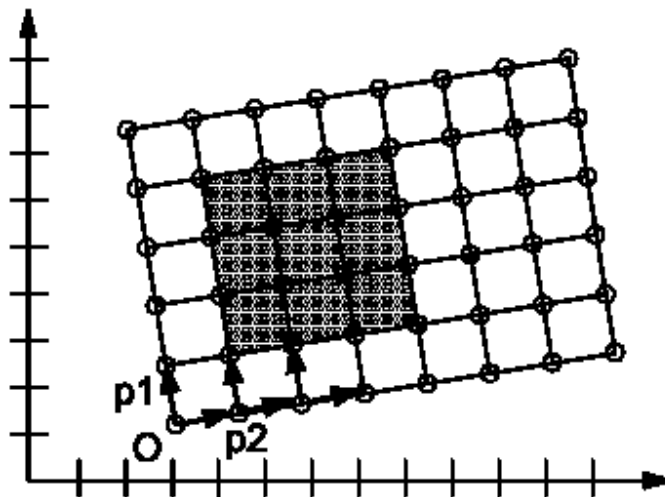
*Приклад 2.* Приклад екземпляру `gml:RectifiedGrid` є таким:

```

<gml:RectifiedGrid dimension="2">
  <gml:limits>
    <gml:GridEnvelope>
      <gml:low>1 1</gml:low>
      <gml:high>4 4</gml:high>
    </gml:GridEnvelope>
  </gml:limits>
  <gml:axisLabels>u v</gml:axisLabels>
  <gml:origin>
    <gml:Point gml:id="palindrome" srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
      <gml:pos>3.1/1</gml:pos>
    </gml:Point>
  </gml:origin>
  <gml:offsetVector srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">0.21.25</gml:offsetVector>
  <gml:offsetVector srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">1.30.2</gml:offsetVector>
</gml:RectifiedGrid>

```

Треба зауважити, що в цьому прикладі трансформована мережа починається з цілочисловим зміщенням 1 1 (значення властивості low) відносно початку, як показано на рис. 6.



**Рисунок 6** — Прив'язана регулярна мережа із ненульовою нижньою межею: p1, p2 – вектори зміщення, O – початок мережі

### 19.3 Схема покриття

#### 19.3.1 AbstractCoverageType, AbstractCoverage

Базовим типом для покриттів є gml:AbstractCoverageType, визначений у схемі таким чином:

```

<complexType name="AbstractCoverageType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:domainSet"/>
        <element ref="gml:rangeSet"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

</sequence>
</extension>
</complexContent>
</complexType>

```

Базові елементи покриття можуть зустрічатися в цій моделі контенту: покриття містить властивості `gml:domainSet` та `gml:rangeSet`. Властивість `gml:domainSet` описує домен покриття, а властивість `gml:rangeSet` описує діапазон покриття.

Абстрактний елемент `gml:AbstractCoverage` реалізує покриття `CV_Coverage` стандарту ISO 19123 (див. Г.2.11 та ISO19123:2005, підрозділ 5.3) та задекларований таким чином:

```

<element name="AbstractCoverage" type="gml:AbstractCoverageType" abstract="true"
substitutionGroup="gml:AbstractFeature"/>

```

Цей елемент виступає як голова групи заміни, яка може містити будь-яке покриття, тип якого походить від `gml:AbstractCoverageType`. Він може діяти як змінна у визначенні моделі контенту, де потрібно дозволити, щоб кожне покриття було дійсним.

### 19.3.2 AbstractDiscreteCoverageType, AbstractDiscreteCoverage

Дискретне покриття складається з доменної множини, діапазону та (необов'язково) функції покриття. Доменна множина складається або з просторових, або з часових об'єктів геометрії, кінцевих у чисельності. Множина діапазону включає кінцеве число значень атрибутів, кожне з яких асоційоване з кожною прямою позицією в межах будь-якого окремого просторово-часового об'єкту у домені. Ця функція покриття перекладає кожний елемент з домену покриття в елемент у його діапазоні. Це визначення відповідає стандарту ISO 19123. Базовий тип для дискретних покриттів `DiscreteCoverageType` визначений у схемі таким чином:

```

<complexType name="DiscreteCoverageType ">
  <complexContent>
    <extension base="gml:AbstractCoverageType">
      <sequence>
        <element ref="gml:coverageFunction" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Елемент функції покриття `coverageFunction` описує переклад “f”, як показано на рисунку 4.

Абстрактний елемент `gml:AbstractDiscreteCoverage` реалізує `CV_DiscreteCoverage` стандарту ISO 19123 (див. Г.2.11 та ISO 19123:2005, підрозділ 5.7) та задекларований таким чином:

```

<element name="AbstractDiscreteCoverage" type="gml:AbstractDiscreteCoverageType"
abstract="true"
substitutionGroup="gml:AbstractCoverage"/>

```

Цей елемент виступає як голова групи заміни, що може містити будь-яке дискретне покриття.

### 19.3.3 AbstractContinuousCoverageType, AbstractContinuousCoverage

Безперервне покриття визначене в ISO 19123 як покриття, що може повертати різні значення для одного і того ж атрибуту географічного об'єкту у різних прямих позиціях у межах одного просторово-часового об'єкту в його просторово-часовому домені. Базовим типом для безперервних покриттів є AbstractContinuousCoverageType, визначений у схемі так:

```
<complexType name="AbstractContinuousCoverageType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractCoverageType">
      <sequence>
        <element ref="gml:coverageFunction" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Елемент coverageFunction описує перехід “f”, як показано на рисунку 5.

Абстрактний елемент gml:AbstractContinuousCoverage задекларований таким чином:

```
<element name="AbstractContinuousCoverage" type="gml:AbstractContinuousCoverageType"
abstract="true"
substitutionGroup="gml:AbstractFeature"/>
```

Цей елемент виступає як голова групи заміни, що може містити будь-яке безперервне покриття, тип якого отримують від gml:AbstractContinuousCoverageType.

### 19.3.4 domainSet, DomainSetType

Елемент властивості gml:domainSet описує просторово-часову область розгляду, в межах якої визначене покриття. Його модель контенту подана через gml:DomainSetType, що визначений таким чином:

```
<element name="domainSet" type="gml:DomainSetType"/>

<complexType name="DomainSetType">
  <sequence minOccurs="0">
    <choice>
      <element ref="gml:AbstractGeometry"/>
      <element ref="gml:AbstractTimeObject"/>
    </choice>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Значення домену є таким чином вибором між gml:AbstractGeometry та gml:AbstractTimeObject. В екземплярі ці абстрактні елементи звичайно будуть замінені комплексом геометрії або часовим комплексом для подання, відповідно, просторових покриттів та часових серій.

**Примітка 1.** Просторово-часові домени підтримуються, якщо домен описаний з використанням складної референційної системи координат, один із компонентів якої є часовим. В іншому випадку, за дотримання комплексу міжнародних стандартів ISO 19100, GML не підтримує комбіновані просторово-часові домени.

Присутність групи `gml:AssociationAttributeGroup` означає, що доменна множина `DomainSet` знаходиться у відповідності до звичайної GML моделі властивості та може використовувати атрибут `xlink:href` для указання на домен як альтернативу включеному опису домену. Семантику власності можна забезпечити з використанням `gml:OwnershipAttributeGroup`.

### 19.3.5 rangeSet, RangeSetType

Елемент властивості `gml:rangeSet` містить значення покриття (які іноді називають атрибутивними значеннями). Його модель контенту подано через `gml:RangeSetType`, що визначають таким чином:

```
<element name="rangeSet" type="gml:RangeSetType"/>

<complexType name="RangeSetType">
  <choice>
    <element ref="gml:ValueArray" maxOccurs="unbounded"/>
    <element ref="gml:AbstractScalarValueList" maxOccurs="unbounded"/>
    <element ref="gml:DataBlock"/>
    <element ref="gml:File"/>
  </choice>
</complexType>
```

Ця модель контенту підтримує структурний опис діапазону. Семантична інформація, що описує набір діапазонів, вбудована з використанням єдиного методу як частина явних значень або як значення шаблону, яке супроводжує подання з використанням `gml:DataBlock` та `gml:File`.

Значення з кожного компоненту (або “каналу”) в діапазоні можна закодувати в елементі `gml:ValueArray` або конкретному члені груп заміни `gml:AbstractScalarValueList`<sup>7)</sup>. Використання цих елементів задовольняє вимогу однорідності типу значення.

### 19.3.6 DataBlock

`gml:DataBlock` описує діапазон (Range) як блок значень, закодованих текстом, подібних до подання у форматі CSV. Модель контенту є такою:

```
<element name="DataBlock" type="gml:DataBlockType" substitutionGroup="gml:AbstractObject"/>

<complexType name="DataBlockType">
  <sequence>
    <element ref="gml:rangeParameters"/>
    <choice>
      <element ref="gml:tupleList"/>
      <element ref="gml:doubleOrNilReasonTupleList"/>
    </choice>
  </sequence>
```

<sup>7)</sup> наприклад, `gml:CategoryList`, `gml:QuantityList` – див. підрозділ 16.4

</complexType>

Параметризація діапазонної множини описана властивістю `gml:rangeParameters`.

### 19.3.7 rangeParameters

Властивість `gml:rangeParameters` описана таким чином:

```
<element name="rangeParameters" type="gml:AssociationRoleType"/>
```

`gml:RangeParameters` є мережею для опису параметрів діапазону. Це може бути локальним описом із використанням придатної схеми запису (дивіться ISO 19103) або він може забезпечувати посилання на зовнішній опис діапазону, що відповідає деякому стандарту. Специфічні параметри діапазону для використання включеними можна визначити шляхом створення GML прикладної схеми, яка може бути базована на схемі об'єктів значень, як описано в підрозділі 16.4.

### 19.3.8 tupleList

Властивість `gml:tupleList` задекларована таким чином:

```
<element name="tupleList" type="gml:CoordinatesType"/>
```

`gml:CoordinatesType` описаний у підпункті 9.1.4.5. Він складається зі списку кортежів координат, в якому кожний кортеж координат відокремлений з допомогою `ts` або розділювального знаку кортежу (пробілу), а кожна координата в кортежі – з допомогою `cs` або розділювального знаку для координат (кома).

Кодування `gml:tupleList` має ефективно переплетені канали.

*Приклад 1.* Набір пар спостережених значень температури та тиску можна записати в `gml:DataBlock` таким чином:

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponents>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:valueComponents>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:tupleList>3,101.2 5,101.3 7,101.4 11,101.5 13,101.6 17,101.7 19,101.7 23,101.8
  29,101.9 31,102.0 37,102.1 41,102.2 43,102.3 47,102.4 53,102.5 59,102.6</gml:tupleList>
</gml:DataBlock>
```

де `Temperature` та `Pressure` є елементами, визначеними в локальній прикладній схемі з використанням `gml:MeasureOrNilReasonListType`.

### 19.3.9 doubleOrNilReasonTupleList

Властивість `gml:doubleOrNilReasonTupleList` задекларована таким чином:

```
<element name="doubleOrNilReasonTupleList" type="gml:doubleOrNilReasonList"/>
```

`gml:doubleOrNilReasonList` описано у підпункті 8.2.4.1 та складається із списку значень `gml:doubleOrNilReason`, кожне з яких відокремлене пробілом. Значення `gml:doubleOrNilReason` згруповані в кортежі, де розмірність кожного кортежу у списку дорівнює числу параметрів діапазону.



*Приклад 1.* Приклад використання `gml:doubleOrNilReasonTupleList` для запису тієї ж самої пари спостережених значень температури та тиску подано у вищенаведеному прикладі в `gml:DataBlock` таким чином:

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponents>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:valueComponents>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:doubleOrNilReasonTupleList>3 101.2 5 101.3 7 101.4 11 101.5 13 101.6 17 101.7
19 101.7 23 101.8 29 101.9 31 102.0 37 102.1 41 102.2 43 102.3 47 102.4 53 102.5 59
102.6</gml:doubleOrNilReasonTupleList>
</gml:DataBlock>
```

### 19.3.10 File, FileType

З метою підвищення ефективності, у GML також передбачені засоби кодування діапазонної множини у довільному зовнішньому коді, такому як бінарний файл. Це кодування може бути широко відомим, але не обов'язковим. У цьому режимі використовують елемент `gml:File`, задекларований таким чином:

```
<element name="File" type="gml:FileType" substitutionGroup="gml:AbstractObject"/>
<complexType name="FileType">
  <sequence>
    <element ref="gml:rangeParameters"/>
    <choice>
      <element name="fileName" type="anyURI"/>
      <element name="fileReference" type="anyURI"/>
    </choice>
    <element name="fileStructure" type="gml:Code IType"/>
    <element name="mimeType" type="anyURI" minOccurs="0"/>
    <element name="compression" type="anyURI" minOccurs="0"/>
  </sequence>
</complexType>
```

У цій версії кодування покриття значення покриття (значення атрибутів у наборі діапазонів) передані в зовнішній файл, на який спрямоване посилання з XML структури, описаної з допомогою `gml:FileType`. Посилання на зовнішній файл спрямоване з допомогою властивості `gml:fileReference`, що є `anyURI` (властивість `gml:filename` скасована). Це означає, що зовнішній файл можна розмістити віддалено від GML екземпляру, який посилається на нього.

*Приклад 1.* Це можна підтримувати, наприклад, як `http`-посилання, так і `SOAP`-вкладення.

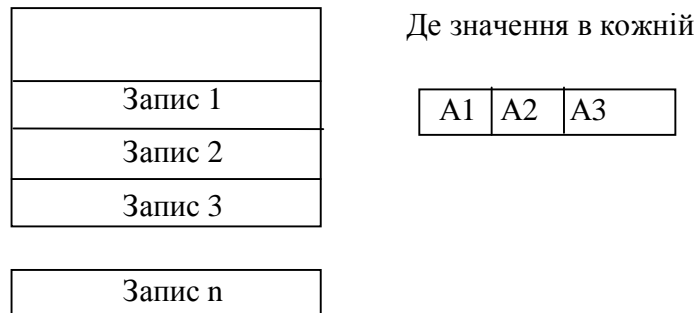
Властивість `gml:compression` вказує на визначення алгоритму стиснення через `anyURI`. Це може бути відновлюване обчислюване визначення чи просто посилання на однозначне ім'я для методу стиснення.

Властивість `gml:mimeType` вказує на визначення `mime` типу файлу.

Властивість `gml:fileStructure` визначена списком кодів. Прикладом значень у списку кодів може бути “переплетений запис” (“Record Interleaved”). Треба зауважити, що всі значення мають бути включені в один файл. Багатофайлові структури не підтримуються в GML.

Семантика діапазонної множини описана вище з використанням властивості `gml:rangeParameters`.

Структура файлу, на який спрямоване посилання, має бути такою, як на рис. 7.



**Рисунок 7** – Структура записів файлу або файл покриття

Треба зауважити, що якщо застосовують будь-який алгоритм стиснення, вищеописану структуру використовують тільки для структур перед стисненням або після декомпресії файлу.

Треба також відзначити, що поля в записі мають відповідати `gml:valueComponents` з `gml:CompositeValue` у порядку в документі.

*Приклад 1.* Кодування бінарного файлу може мати такий вигляд:

```
<gml:File>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponents>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:valueComponents>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:fileName>http://www.somedata.org/temp\_pressure.dat</gml:fileName>
  <gml:fileStructure>Record Interleaved</gml:fileStructure>
</gml:File>
```

### 19.3.11 coverageFunction, CoverageFunctionType

У цьому пункті описана властивість `gml:coverageFunction` тобто, перехід від “Г” (див. рисунок 4) з домену до діапазону покриття. Модель контенту для функції покриття подано таким чином:

```
<element name="coverageFunction" type="gml:CoverageFunctionType"
substitutionGroup="gml:AbstractObject"/>
```

```
<complexType name="CoverageFunctionType">
```

```

<choice>
  <element ref="gml:MappingRule"/>
  <element ref="gml:CoverageMappingRule">
  <element ref="gml:GridFunction"/>
</choice>
</complexType>

```

Треба зауважити, що значення функції покриття CoverageFunction є одним із *gml:MappingRule* (скасоване), *gml:CoverageMappingRule* та *gml:GridFunction*.

Якщо властивість *gml:coverageFunction* пропущено у випадку покриття регулярної мережі (включно із трансформованими покриттями регулярної мережі), то за значення властивості *gml:low* у геометрії *gml:Grid* приймають *gml:startPoint*, та припускають, що *gml:sequenceRule* є лінійним, а властивість *gml:axisOrder* становить "+1 +2".

*Приклад 1.* Ці значення за умовчанням найкраще можна показати на такому простому прикладі:

```

<AverageTempPressure
xmlns="http://www.opengis.net/app" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app ./CoverageExamples.xsd">
  <gml:domainSet>
    <gml:Grid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>4 4</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>x y</gml:axisLabels>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponents>
            <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
            <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
          </gml:valueComponents>
        </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:tupleList>3,101.2 5,101.3 7,101.4 11,101.5 13,101.6 17,101.7 19,101.7 23,101.8
29,101.9 31,102.0 37,102.1 41,102.2 43,102.3 47,102.4 53,102.5 59,102.6</gml:tupleList>
    </gml:DataBlock>
  </gml:rangeSet>
</AverageTempPressure>

```

Оскільки жодна *coverageFunction* не визначена, припускають, що ця функцію є функцією лінійного сканування з початком порядку "+1 +2" в позиції (0 0). Якщо подивитися на *DataBlock*, можна побачити, що є перехід, показаний у таблиці 4:

Таблиця 7 – Приклад блоку даних

Позиція в регулярній мережі	Значення даних
00	3,101.2
10	5,101.3
20	7,101.4
30	11,101.5
01	13,101.6
11	17,101.7
21	19,101.7
31	23,101.8
02	29,101.9
12	31,102.0
22	37,102.1
32	41,102.2
03	43,102.3
13	47,102.4
23	53,102.5
33	59,102.6

### 19.3.12 CoverageMappingRule, MappingRule

`gml:CoverageMappingRule` забезпечує формальний чи неформальний опис функції покриття:

```
<element name="CoverageMappingRule" type="gml:MappingRuleType"
substitutionGroup="gml:AbstractObject"/>
```

```
<complexType name="MappingRuleType" final="#all">
  <choice>
    <element name="ruleDefinition" type="string"/>
    <element name="ruleReference" type="gml:ReferenceType"/>
  </choice>
</complexType>
```

*Приклад 1.* Можна використовувати MathML для формальних описів, а неформальні можуть бути будь-яким вільним текстом.

Правило переходу можна визначити як включений рядок (`gml:ruleDefinition`) або через віддалене посилання з допомогою `xlink:href` (`gml:ruleReference`).

Якщо жодне ім'я правила не визначене, значенням за умовчанням є 'Linear' відносно членів домену у порядку документу.

*gml:CoverageMappingRule є заміною для скасованого gml:MappingRule:*

### 19.3.13 GridFunction, GridFunctionType

`gml:GridFunction` забезпечує явним чином правило переходу для геометрій регулярної мережі, тобто домен має бути геометрією типу регулярної мережі. Воно описує перехід контрольних точок у мережі (дискретне регулярне точкове покриття) чи комірок мережі (дискретне покриття поверхні) у значення в діапазонній множині. Модель контенту є такою:

```
<element name="GridFunction" type="gml:GridFunctionType"
substitutionGroup="gml:AbstractObject"/>
```

```
<complexType name="GridFunctionType">
  <sequence>
    <element name="sequenceRule" type="gml:SequenceRuleType" minOccurs="0"/>
    <element name="startPoint" type="gml:integerList" minOccurs="0"/>
  </sequence>
</complexType>
```

`gml:startPoint` є індексною позицією точки в регулярній мережі, яка переходить у першу точку в діапазонній множині (це є також індексною позицією першої контрольної точки мережі). Якщо властивість `gml:startPoint` property пропущено, то приймають `gml:startPoint` рівним значенню `gml:low` в геометрії `gml:Grid`. Наступні точки в переході визначені значенням `gml:sequenceRule`.

#### 19.3.14 `sequenceRule`, `SequenceRuleType`, `SequenceRuleEnumeration`

Правило послідовності `sequenceRule` визначене в моделі контенту:

```
<complexType name="SequenceRuleType">
  <simpleContent>
    <extension base="gml:SequenceRuleNames">
      <attribute name="order" type="gml:IncrementOrder"/>
      <attribute name="axisOrder" type="gml:AxisDirectionList"/>
    </extension>
  </simpleContent>
</complexType>
```

`gml:SequenceRuleType` отримують від `gml:SequenceRuleEnumeration` через додання атрибуту `axisOrder`. `gml:SequenceRuleEnumeration` є переліченим типом, визначеним таким чином:

```
<simpleType name="SequenceRuleEnumeration">
  <restriction base="string">
    <enumeration value="Linear"/>
    <enumeration value="Boustrophedonic"/>
    <enumeration value="Cantor-diagonal"/>
    <enumeration value="Spiral"/>
    <enumeration value="Morton"/>
    <enumeration value="Hilbert"/>
  </restriction>
</simpleType>
```

Ці імена правил визначені в ISO 19123.

Якщо жодне правило не визначене, то значенням за умовчанням є “*Linear*”.

Атрибут `axisOrder` має таку модель контенту:

```
<simpleType name="AxisDirectionList">
  <list itemType="gml:AxisDirection"/>
</simpleType>
```

```
<simpleType name="AxisDirection">
  <restriction base="string">
    <pattern value="[\+|-][1-9][0-9]*"/>
  </restriction>
</simpleType>
```

Значення `gml:AxisDirection` вказує на порядок зростання, який треба використовувати на осях мережі.

*Приклад 1.* "+3" означає, що треба проходити точки в мережі з найнижчих до найвищих на третій осі.

Різні значення у `gml:AxisDirectionList` вказують на порядок зростання, який треба використовувати на всіх осях мережі. Кожну вісь треба згадувати один і тільки один раз.

*Приклад 2.* "+1 -2 +3" означає, що треба проходити точки з найнижчої до найвищої на першій на вісі, починаючи з точок з найвищим значенням на другій осі та найнижчим значенням на третій вісі, з прирощенням, найшвидшим на першій вісі, далі з прирощенням на другій вісі і, врешті, на третій вісі.

### 19.3.15 Особливі типи покриття у GML

GML підтримує типи дискретних покриттів, визначені в ISO 19123.

Типи, що підтримуються, можна використовувати, замінюючи ними `gml:AbstractDiscreteCoverage`, та включають:

- `gml:MultiPointCoverage` (`CV_DiscretePointCoverage`)
- `gml:MultiCurveCoverage` (`CV_DiscreteCurveCoverage`)
- `gml:MultiSurfaceCoverage` (`CV_DiscreteSurfaceCoverage`)
- `gml:MultiSolidCoverage` (`CV_DiscreteSolidCoverage`)
- `gml:GriddedCoverage` (`CV_DiscreteGridPointCoverage`)
- `gml:RectifiedGridCoverage` (`CV_DiscreteGridPointCoverage`)

**Примітка 1.** Конкретні типи безперервних покриттів можна очікувати в подальших випусках цього Міжнародного стандарту.

Користувачі можуть також будувати їх власні типи покриттів через отримання від `gml:DiscreteCoverageType`, `gml:AbstractContinuousCoverageType` або через отримання з вищезгаданих особливих типів покриттів.

Такі ж кодування множини діапазону можна застосовувати до кожного конкретного типу дискретних покриттів, згідно до визначення останніх геометричним типом домену.

### 19.3.16 MultiPointCoverage

У `gml:MultiPointCoverage` доменною множиною є `gml:MultiPoint`, що є колекцією довільно розподілених геометричних точок. Тобто, значення у `gml:domainSet` мають становити `gml:MultiPoint`.

```
<element name="MultiPointCoverage" type="gml:DiscreteCoverageType"
  substitutionGroup="gml:AbstractDiscreteCoverage"/>
```

У `gml:MultiPointCoverage` перехід від домену до діапазону є прямим:

- для кодів `gml:DataBlock` точки, що належать до `gml:MultiPoint`, переходять у порядку документу в кортежі блоку даних;
- для кодів `gml:CompositeValue` точки, що належать до `gml:MultiPoint`, переходять у члени композитного значення в порядку документу;

– для кодів gml:File точки, що належать до gml:MultiPoint, переходять у записи файлу у послідовному порядку.

*Приклад 1.* У gml:MultiPointCoverage використовують кодування значень:

```

<AverageTempPressure
  xmlns="http://www.opengis.net/app"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/app CoverageExamples.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
      <gml:lowerCorner>1 1</gml:lowerCorner>
      <gml:upperCorner>4 4</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:DomainSet>
    <gml:MultiPoint srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <gml:pointMember>
        <gml:Point>
          <gml:pos>1 1</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point>
          <gml:pos>2 2</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point>
          <gml:pos>3 3</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point>
          <gml:pos>4 4</gml:pos>
        </gml:Point>
      </gml:pointMember>
    </gml:MultiPoint>
  </gml:DomainSet>
  <gml:rangeSet>
    <gml:ValueArray>
      <gml:valueComponents>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">3</Temperature>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">5</Temperature>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">7</Temperature>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">11</Temperature>
      </gml:valueComponents>
    </gml:ValueArray>
  </gml:rangeSet>

```

</AverageTempPressure>

### 19.3.17 MultiCurveCoverage

У gml:MultiCurveCoverage домен поділено на колекцію ліній, що складають gml:MultiCurve. Функція покриття далі перекладає кожну лінію в колекції у значення в множині діапазону.

Модель контенту для gml:MultiCurveCoverage є такою:

```
<element name="MultiCurveCoverage" type="gml:DiscreteCoverageType "
substitutionGroup="gml:AbstractDiscreteCoverage"/>
```

Значення gml:DomainSet має становити gml:MultiCurve.

У gml:MultiCurveCoverage перехід від домену до діапазону є прямим:

- для кодувань gml:DataBlock лінії, що належать до gml:MultiCurve, переходять у порядку документу у кортежі блоку даних;
- для кодувань gml:CompositeValue лінії, що належать до gml:MultiCurve, переходять у члени композитного значення в порядку документу;
- для кодувань gml:File лінії, що належать до gml:MultiCurve переходять, у записи файлу у послідовному порядку.

*Приклад 1.* gml:MultiCurveCoverage із використанням кодувань блоку даних:

```
<AverageTempPressure
  xmlns="http://www.opengis.net/app"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/app CoverageExamples.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
      <gml:lowerCorner>1.1 1.1</gml:lowerCorner>
      <gml:upperCorner>5.5 5.5</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:multiCurveDomain>
    <gml:MultiCurve srsName="urn: x-ogc:def:crs:EPSG:6.6:4326">
      <gml:curveMember>
        <gml:LineString>
          <gml:posList dimension="2">1.1 1.1 2.2 2.2</gml:posList>
        </gml:LineString>
      </gml:curveMember>
      <gml:curveMember>
        <gml:LineString>
          <gml:posList dimension="2">2.2 2.2 3.3 3.3</gml:posList>
        </gml:LineString>
      </gml:curveMember>
      <gml:curveMember>
        <gml:LineString>
          <gml:posList dimension="2">3.3 3.3 4.4 4.4</gml:posList>
        </gml:LineString>
      </gml:curveMember>
      <gml:curveMember>
```



```

    <gml:LineString>
      <gml:posList dimension="2">4.4 4.4 5.5 5.5</gml:posList>
    </gml:LineString>
  </gml:curveMember>
</gml:MultiCurve>
</gml:domainSet >
<gml:rangeSet>
  <gml:DataBlock>
    <gml:rangeParameters>
      <gml:CompositeValue>
        <gml:valueComponents>
          <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
          <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
        </gml:valueComponents>
      </gml:CompositeValue>
    </gml:rangeParameters>
    <gml:doubleOrNilReasonTupleList>3 101.25 101.37 101.4 11 101.5</gml:doubleOrNilReasonTupleList>
  </gml:DataBlock>
</gml:rangeSet>
</AverageTempPressure>

```

### 19.3.18 MultiSurfaceCoverage

У `gml:MultiSurfaceCoverage` домен поділений на колекції поверхонь, що становлять `gml:MultiSurface`. Функція покриття далі перекладає кожену поверхню в колекції в значення у діапазонній множині.

```

<element name="MultiSurfaceCoverage" type="gml:DiscreteCoverageType"
  substitutionGroup="gml:AbstractDiscreteCoverage"/>

```

Модель контенту для `gml:MultiSurfaceCoverage` має таку форму:

```

<element name="MultiSurfaceCoverage" type="gml:MultiSurfaceCoverageType"
  substitutionGroup="gml:AbstractDiscreteCoverage"/>

```

Значення в `gml:domainSet` має становити `gml:MultiSurface`.

У `gml:MultiSurfaceCoverage` перехід від домену до діапазону є прямим:

- для кодувань `gml:DataBlock` поверхні, що належать до `gml:MultiSurface`, переходять у порядку документу в кортежі блоку даних;
- для кодувань `gml:CompositeValue` поверхні, що належать до `gml:MultiSurface`, переходять у члени композитного значення у порядку документу;
- для кодувань `gml:File` поверхні, що належать до `gml:MultiSurface`, переходять у записи файлу у послідовному порядку.

*Приклад 1.* У `gml:MultiSurfaceCoverage` використовують таке кодування в файлі:

```

<SoilData xmlns="http://www.opengis.net/app" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/app
./CoverageExamples.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">
      <gml:lowerCorner>1 2 3</gml:lowerCorner>
    </gml:Envelope>
  </gml:boundedBy>

```

```

    <gml:upperCorner>10 11 12</gml:upperCorner>
  </gml:Envelope>
</gml:boundedBy>
<gml:domainSet>
  <gml:MultiSurface srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">
    <gml:surfaceMember>
      <gml:Polygon gid="p1">
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList dimension="3">1 1 1 1 5 1 5 5 1 5 1 1 1 1 1</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
    <gml:surfaceMember>
      <gml:Polygon gid="p6">
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList dimension="3">10 1 2 5 1 1 5 5 1 10 5 2 10 1 2</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
    <gml:surfaceMember>
      <gml:Polygon gid="p11">
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList dimension="3">5 5 1 1 5 1 1 10 1 5 10 1 5 5 1</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
    <gml:surfaceMember>
      <gml:Polygon gid="p16">
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList dimension="3">10 5 2 5 5 1 5 10 1 10 10 2 10 5 2</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
  </gml:MultiSurface>
</gml:domainSet>
<gml:rangeSet>
  <gml:File>
    <gml:rangeParameters>
      <gml:Composite Value>
        <gml:valueComponent>
          <SoilType codeSpace="http://my.big.org/classifications/soils">template</SoilType>
        </gml:valueComponent>
      </gml:Composite Value>
    </gml:rangeParameters>
  </gml:File>
</gml:rangeSet>

```

```

    </gml:valueComponent>
    <gml:valueComponent>
      <SoilMoisture uom="http://my.big.org/units/percent">template</SoilMoisture>
    </gml:valueComponent>
  </gml:CompositeValue>
</gml:rangeParameters>
<gml:fileReference>soil.dat</gml:fileReference>
<gml:fileStructure>Record Interleaved</gml:fileStructure>
</gml:File>
</gml:rangeSet>

```

### 19.3.19 MultiSolidCoverage

У `gml:MultiSolidCoverage` домен поділений на колекцію тіл, що включає `gml:MultiSolid`. Функція покриття перекладає кожне тіло в колекції у значення в діапазонній множині.

```

<element name="MultiSolidCoverage" type="gml:DiscreteCoverageType"
substitutionGroup="gml:AbstractDiscreteCoverage"/>

```

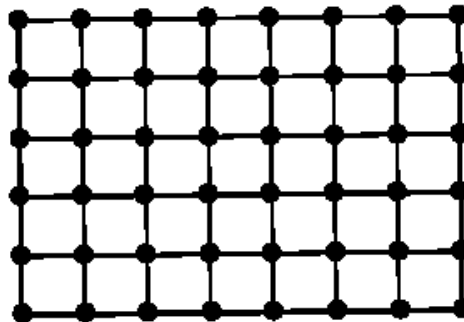
Значення в `gml:domainSet` має становити `gml:MultiSolid`.

У `gml:MultiSolidCoverage` перехід від домену до діапазону є прямим:

- для кодувань `gml:DataBlock` тіла, що належать до `gml:MultiSolid` переходять у порядку документу в кортежі блоку даних;
- для кодувань `gml:CompositeValue` тіла, що належать до `gml:MultiSolid`, переходять у члени композитного значення у порядку документу;
- для кодувань `gml:File` тіла, що належать до `gml:MultiSolid`, переходять у записи файлу у послідовному порядку.

### 19.3.20 GridCoverage

`gml:GridCoverage` є дискретним точковим покриттям, у якому множина домену є геометричною регулярною мережею (GRID) точок (див. рисунок 8).



**Рисунок 8** – Домен покриття регулярної мережі є регулярною мережею точок

Модель контенту для покриття регулярної мережі точок є такою:

```

<element name="GridCoverage" type="gml:DiscreteCoverageType"
substitutionGroup="gml:AbstractDiscreteCoverage"/>

```

Треба зауважити, що це те ж саме, що і `gml:MultiPointCoverage`, за виключенням того, що значення в `gml:domainSet` має становити `gml:Grid`.

`gml:Grid` визначений у пункті 19.2.2. Треба зауважити, що просте регулярне покриття не є геометрично прив'язаним і тому не має геометричних позицій, які можна призначити точкам регулярної мережі. Таке геометричне позиціювання реалізовано в `gml:RectifiedGridCoverage`, розглянутому в пункті 19.3.21.

**Примітка 1.** Коли точку регулярної мережі використовують для подання пробного простору, то точка регулярної мережі являє центр пробного простору, див. 19.2.2

*Приклад 1.* У `gml:GridCoverage` використовують файлове кодування для його значень:

```
<AverageTempPressure>
  <gml:domainSet>
    <gml:Grid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>4 4</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>x y</gml:axisLabels>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:File>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponents>
            <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
            <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
          </gml:valueComponents>
        </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:fileReference>http://www.somedata.org/temp\_pressure.dat</gml:fileReference>
      <gml:fileStructure>Record Interleaved</gml:fileStructure>
    </gml:File>
  </gml:rangeSet>
</AverageTempPressure>
```

### 19.3.21 RectifiedGridCoverage

`gml:RectifiedGridCoverage` є дискретним точковим покриттям, базованим на трансформований регулярній мережі, подібним до покриття регулярної мережі у пункті 19.3.20, за виключенням того, що точки мережі геометрично прив'язані. Трансформоване покриття регулярної мережі має домен, що є геометрією `gml:RectifiedGrid`, як визначено в пункті 19.2.3.

Модель контенту для `gml:RectifiedGridCoverage` є такою:

```
<element name="RectifiedGridCoverage" type="gml:DiscreteGridCoverageType"
  substitutionGroup="gml:AbstractDiscreteCoverage"/>
```

Значення в `gml:domainSet` має становити `gml:RectifiedGrid`.

`gml:RectifiedGrid` визначено в пункті 19.2.3.

*Приклад 1.* `gml:RectifiedGridCoverage` (з використанням блоку даних):

```

<AveragePressure xmlns="http://www.opengis.net/app" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/app
./CoverageExamples.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">
      <gml:lowerCorner>1.2 3.3 2.1</gml:lowerCorner>
      <gml:upperCorner>13.6 12.1 15.3</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:domainSet>
    <gml:RectifiedGrid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>1 1</gml:low>
          <gml:high>4 4</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>u v</gml:axisLabels>
      <gml:origin>
        <gml:Point gml:id="palindrome" srsName="urn:x-ogc:def:crs:EPSG:6.6:4329">
          <gml:pos>1.2 3.3 2.1</gml:pos>
        </gml:Point>
      </gml:origin>
      <gml:offsetVector srsName="urn:ogc:def:crs:EPSG:6.6:4329">1.1 2.2 3.3</gml:offsetVector>
      <gml:offsetVector srsName="urn:ogc:def:crs:EPSG:6.6:4329">2.0 1.0 0.0</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:rangeParameters>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:rangeParameters>
      <gml:doubleOrNilReasonTupleList>101.2 101.3 101.4 101.5 101.6 101.7 101.7 101.8
101.9 102.0 102.1 102.2 102.3 102.4 102.5 102.6</gml:doubleOrNilReasonTupleList>
    </gml:DataBlock>
  </gml:rangeSet>
</AveragePressure>

```

## 20 ПРОФІЛІ

### 20.1 GML профілі та прикладні схеми

GML є комплексним стандартом, що є багатим та виразним. Взагалі, аплікація не потребує використання цілої GML, але може задіяти підмножину конструктів, що відповідають спеціальним доречними вимогами.

Треба використовувати таке визначення профілю (ISO/IEC TR 10000-1:1998 та ISO 19106:2004):

“Профіль – це набір одного чи більше базових стандартів та [профілів], та, де це застосовне, ідентифікацій вибраних класів [(типів, атрибутів та елементів)], відповідних

підмножин, опцій та параметрів цих базових стандартів або [профілів], необхідних для здійснення окремої функції”.

Це було визначено для моделі архітектури OSI, тому треба перекладати “клас” в “типи, атрибути та елементи” для застосування цього визначення до XML схеми. Є кілька шляхів реалізації цього, і у профілях GML використовують підхід “копіювати та видаляти”. Для створення профілю розробник може копіювати застосовні файли схеми та просто видаляти будь-які глобальні типи, елементи та локальні необов’язкові частки, яких він не потребує для своєї прикладної схеми.

### 20.2 Визначення профілю

Профіль GML можна визначити для підсилення інтеоперабельності та зняття неоднозначності через дозвіл тільки специфічного піднабору GML. Прикладні схеми далі можуть бути узгодженими із таким профілем для того, щоб використати перевагу з будь-яких покращень інтеоперабельності та продуктивності, які він забезпечує у порівнянні з повною GML. Такі профілі можна визначити для прикладних схем, що включені в інші специфікації.

Трапляються випадки, коли скорочена функціональність прийнятна, або коли вимоги до обробки примушують використовувати логічний піднабір GML. Наприклад, аплікації, що не вимагають маніпуляцій з атрибутами XLink у будь-якій формі, можуть відповідати специфічному профілю, що виключає їх; обмеженням у цьому випадку буде невикористання посилань. Інші випадки можуть включати визначальні обмеження на рівень вкладеної структури в тегах (тобто, глибини дерева), або тільки дозволяючи наявність географічних об’єктів з однорідними властивостями як членів колекції географічних об’єктів. Часто такі обмеження можуть бути введені через нові схеми; інші можна ввести через процедурні домовленості в рамках інформаційної спільноти.

### 20.3 Зв’язок з прикладною схемою

Профіль може бути на початку прикладної схеми.

*Приклад 1.* Профіль сервісу, базований на розміщенні, може обмежувати типи геометрії до тих, які використовують у застосуваннях у LBS аплікаціях, та LBS прикладні схеми можуть далі додавати елементи “PointCircle,” “PointEllipse” та “PointArc” для реалізації елементів LIF “CIRCLE,” “ELLIPSE” та “ARC”, які використовують для опису обрахунку помилок розташування мобільного пристрою.

Побудова таких прикладних схем є, таким чином, процесом, що складається з двох частин. Профіль діє як обмеження GML для створення типів та елементів, узгоджених із повною GML, але з потенційним браком деяких необов’язкових часток. Далі у прикладній схемі використовують ці типи як загальну базу та використовують їх у нових типах та елементах шляхом розширення та включення.

GML → відбір & обмеження → GML профіль → розширення & включення → прикладна схема.

### 20.4 Правила для елементів та типів профілю

Глобальні профільовані елементи у профілі GML повинні:

- поділяти одне й те ж ім’я або простір імен батьківського елемента у GML;
- включати всі обов’язкові частки (піделементи та атрибути) батьківського елемента в GML;
- не включати частки, яких немає в батьківському елементі у GML;

- мати ті ж значення за умовчанням для атрибутів, що і батьківський елемент у GML;
- мати паралельну ієрархію груп заміни для іменованих елементів у обох схемах. Глобальні типи у профілі GML повинні:
- поділяти те ж саме ім'я або простір імен, що і батьківський тип у GML;
- включати всі обов'язкові частки (піделементи та атрибути) батьківського типу в GML;
- не включати частки, яких немає в батьківському типі в GM;
- мати ті ж значення за умовчанням для атрибутів, що і батьківський тип у GML;
- мати паралельне дерево походження для іменованих типів в обох схемах; Документи екземплярів профілю повинні бути дійсними для повної GML схеми. Використовуючи метафору “копіювати та видаляти”, описану вище, розробник

може:

- видаляти глобальні елементи та глобальні типи;
- видаляти необов'язкові піделементи з будь-яких типів або елементів;
- робити необов'язкові піделементи та атрибути обов'язковими в будь-якому типі елементів (якщо значення за умовчанням існує, воно має бути вилучене, або перевірка схеми сформує звіт про помилки – значення за умовчанням дійсні тільки для необов'язкових часток);
- обмежувати потужність множини будь-якої частки.

Ніщо з перерахованого не може впливати на дійсність документу, що розроблений за профілем, але перевірений за повною GML схемою.

Розробник не може:

- видаляти обов'язкові піделементи з будь-яких типів та елементів;
- робити обов'язкові частки необов'язковими;
- послаблювати обмеження потужності множини для будь-якої частки;
- додавати або змінювати значення за умовчанням чи фіксоване значення.

Останній пункт є більш складним ніж інші. Документи, дійсні за профілем, будуть дійсними і за повної GML схеми, але інтерпретація цих документів зміниться. Наприклад, якщо профіль визначає, що референційною системою координат має бути UTM, та повна схема визначає WGS 84 (широта, довгота) як референційну систему координат за умовчанням, тоді інтерпретація файлу зміниться при переході від профілю до повної схеми.

### **20.5 Правила посилань на GML профілі з прикладних схем**

GML прикладна схема має посилатися на повну GML схему в атрибуті `schemaLocation` елементу `<import>`.

Документ GML прикладної схеми, що відповідає одному чи більше GML профілям, має надавати елемент `<gml:ProfileSchema>` для кожного профілю кореневого документу схеми `<schema>`, де значенням є розташування схеми профілю. Треба відзначити, що прикладна схема може узгоджуватись із багатьма профілями.

Приклад:

```
<schema ...>
  <annotation>
    <appInfo>
```

```

    <gml:gmlProfileSchema>http://schemas.opengis.net/gml/3.2.1/profiles/gmlSimpleFeatureProfile/1.1.0/gmlsf.xsd</gml:gmlProfileSchema>
    <gml:gmlProfileSchema>http://schemas.opengis.net/gml/3.2.1/profiles/gmlPointProfile/1.1.0/gmlPointProfile.xsd</gml:gmlProfileSchema>
  </appInfo>
</annotation>
...
</schema>

```

Елемент <gml: gml ProfileSchema> визначений таким чином:

```
<element name="gmlProfileSchema" type="anyURI"/>
```

### 20.6 Рекомендації для прикладних схем з використанням GML профілів

Для того, щоб профіль в прикладній схемі можна було пізніше розширити для включення інших профільованих елементів GML, розроблені такі рекомендації:

- глобальні елементи, яких немає в профілі GML, але є в прикладній схемі з використанням профілю GML, не повинні мати те ж саме ім'я, що і будь-якій елемент у GML схемі;
- глобальні типи, яких немає яких немає в профілі GML, але є в прикладній схемі з використанням профілю GML, не повинні мати те ж саме ім'я, що і будь-який тип у GML схемі.

Якщо тип або елемент у прикладній схемі призначений для універсального використання, то подані вище умови допоможуть прикладній схемі в міграції цього типу або елемента від його власного простору імен до такого, як визначено в GML.

Наступні рекомендації подані лише для зручності для тих, хто намагається зрозуміти роль профілю в прикладній схемі:

- профільовані елементи та типи треба включати або в один файл для меншого профілю, або в файлову структуру, яка паралельна такій же, що притаманна GML; домовленість про точне іменування цього паралелізму покладена на автора прикладної схеми;
- посилання на відповідний документ GML прикладної схеми треба подавати в коментарі біля початку файлу;

**Примітка 1.** Метод, що показав свою зручність – упаковування потрібних GML компонентів у документ схеми, що називається, наприклад, “gmlForApplicationDomain.xsd”. Цей документ може містити копію необхідних елементів, зібраних у досконалий спосіб (наприклад, дивіться додаток Ж), або можуть просто <включати> піднабір документів схеми, що охоплює стандартний GML розподіл. Документ схеми gml.xsd – це вичерпний наднабір згідно до останнього підходу.

### 20.7 Підсумок правил для GML профілів

В цілому, правила для профілю є такими:

- профіль GML є логічним обмеженням піднабору GML;
- профіль не може змінювати ім'я, визначення або тип даних обов'язкових елементів чи атрибутів GML;



- доречна схема чи схеми, які визначають профіль в ядрі ‘gml’простору імен, знаходиться за адресою <http://www.opengis.net/gml/3.2>;
- в прикладній схемі може розширювати та використовувати типи з профілю, але треба робити це в її власному просторі імен та не використовувати <http://www.opengis.net/gml/3.2>.

Функціональним тестом на дотримання цих правил є:

Ситуація, коли будь-який документ екземпляру для прикладної схеми з використанням профілю GML буде дійсним за тією ж самою прикладною схемою, якщо замінити профіль GML повною GML схемою. Далі, інтерпретація цього документі не зміниться, незалежно від того, яку з двох схем використовували.

## 21 ПРАВИЛА ДЛЯ GML ПРИКЛАДНИХ СХЕМ

### 21.1 GML екземпляри об’єктів

#### 21.1.1 GML документи

XML документ містить один XML елемент як свій корінь. GML документ може бути одним із з таких елементів:

- gml:AbstractFeature чи будь-який елемент, прямо чи непрямо присутній у його групі заміни;
- gml:Dictionary чи будь-який елемент, прямо чи непрямо присутній у його групі заміни (це включає колекції об’єктів та покриття, що є також географічними об’єктами);
- gml:TopoComplex чи будь-який елемент, прямо чи непрямо присутній у його групі заміни.

**Примітка 1.** gml:Dictionary включає словники референцих систем координат та одиниць.

Стандартні методи для XML документів, базованих на W3C XML схемі, забезпечують декларування XML просторів імен, використаних у документі, як атрибутів у документі, та розміщення документів схеми, що забезпечує указання на вихідні компоненти для кожного простору імен.

Для GML документу джерелом компонентів, що описують первинні компоненти в документі, є GML прикладна схема. І тип документу, і асоційована GML прикладна схема описані в цьому розділі.

Слід зауважити, що це не означає, що всі елементи та атрибути в GML документі визначені однією GML прикладною схемою. Компоненти схеми, на які посилається GML документ, можуть міститись у будь-якому числі GML прикладних схем або інших XML схем.

#### 21.1.2 GML елементи об’єкту в інших XML документах

GML елементи об’єктів можуть зустрічатися і в XML документах, що не є GML документами. XML документ має бути дійсним згідно з XML документом схеми, що імпортує (прямо чи непрямо) GML схему або профіль GML та (необов’язково) одну чи більше GML прикладних схем.

*Приклад 1.* Елементи GML об’єкту можна використовувати в повідомленнях запитів та відповідях Web-служб.

## 21.2 GML прикладні схеми

### 21.2.1 Вступ

GML прикладна схема є XML схемою, у якій дотримані правила, оговорені в цьому розділі, в якому описані один чи більше типів географічних об'єктів, компоненти географічних об'єктів, метадані, включно з словниками та визначеннями, які використовують для визначення географічних об'єктів. GML прикладна схема визначає словник для часткового домену, що розглядається, через визначення та опис термінів цього словника (див. ISO 19109) таким чином:

Прикладна схема може посилатися на пряму на конкретні, глобальні GML елементи (включно з групами) та атрибути (включно з атрибутами груп (attributeGroups)), іменами та моделями контенту яких точно подані компоненти словника, які він визначає.

*Приклад 1.* Це включає елементи властивостей gml:name або gml:description, елементи об'єктів типу gml:Observation, gml:Dictionary, або gml:Point, та атрибути типу gml:id.

У прикладній схемі можна декларувати нові елементи та атрибути в їх власному просторі імен з використанням GML типів, якщо словник, що визначає його, потребує різних імен для одних й тих же моделей контенту для розрізнення їх семантичних ролей. Елемент, задекларований у прикладній схемі, буде в іншому просторі імен та може бути використаний у документі екземпляру.

*Приклад 2.* gml:EnvelopeType можна використовувати в незміненому вигляді як модель контенту для елемента xmlns:Interval.

*Приклад 3.* gml:LengthType можна використовувати в незміненому вигляді як модель контенту для елемента ex:height.

*Приклад 4.* gml:PointPropertyType можна використовувати на пряму як модель контенту для елемента властивості ex:representativePoint.

Прикладна схема може отримувати нові типи в її власному просторі імен через розширення GML типів, коли словник потребує включення компонентів з додатковими властивостями, згідно до специфіки домену.

**Примітка 1.** Визначення типів географічних об'єктів для специфічної аплікації вимагає, щоб модель контенту типів географічних об'єктів була отримана з типу gml:AbstractFeatureType, звичайно через розширення.

*Приклад 5.* Визначення нових типів геометрії не передбачено в GML схемі, але його вимагає аплікація, наприклад, еліпс.

Прикладна схема може отримувати нові типи в її власному просторі імен через обмеження GML типів, коли словник, що визначає його, потребує включення більш спеціалізованих версій GML, які обмежують потужність множини їх властивостей.

*Приклад 6.* Аплікація вимагає заборонити використання множинних імен в їх типах географічного об'єкту. Цього можна досягнути через отримання кореневого типу географічного об'єкту для специфічної аплікації через обмеження від gml:AbstractFeatureType, що встановлює максимальну множинність gml:name, рівною 1.

Прикладна схема може декларувати нові елементи, які призначені в групу заміни, очолювану абстрактним або конкретним GML елементом. Елемент, задекларований у прикладній схемі, може зустрічатися в документах екземплярів на місці голови групи заміни та узгоджуватися з моделлю контенту, яка посилається на голову групи заміни. Слід зауважити, що для того, щоб бути дійсним членом групи заміни, тип елемента має

бути правильно отриманий з типу елемента, що є головою групи заміни. Всі абстрактні елементи в GML схемі тільки тоді корисні, коли діють як голови груп заміни.

*Приклад 7.* gml:AbstractGML, gml:AbstractFeature, gml:AbstractGeometry, gml:AbstractCoverage можуть служити як голови груп заміни для елементів прикладної схеми.

В прикладній схемі можна декларувати нові елементи, атрибути в їх власному просторі імен з допомогою типів, що вона визначила для надання імен специфічних за словником, їх моделям контенту.

*Приклад 8.* Типи даних або переліки, специфічні за аплікацією.

Всі GML прикладні схеми побудовані з використанням загальних правил цього розділу з одного чи більше GML компонентів схеми, визначених у розділах з 6.5 по 19.

GML дозволяє отримувати багато інших видів елементів, таких як нові одиниці вимірювання, нові властивості геометрії та нові геометрії. Тоді як ці елементи можна зібрати в пакети в окремих схемах, їх розглядають як підлеглі щодо категорій схем цього розділу. Будь-яка GML прикладна схема має належати хоча б до одного з типів схеми, описаних у підрозділах 21.3 – 21.11, та відповідати правилам із відповідних підрозділів у додаток до підрозділу 21.2. Тому допускають створення GML прикладної схеми, що визначає географічні об'єкти, покриття та значення, якщо вона відповідає правилам, викладеним у підрозділах 21.2, 21.3, 21.8 та 21.9.

### 21.2.2 Цільовий простір імен

Прикладна схема має декларувати цільовий простір імен. Це є простір імен, у якому терміни для географічних об'єктів та властивостей словника визначені прикладною схемою live. Це не має бути GML простір імен (<http://www.opengis.net/gml/3.2>). Є домовленість, що ідентифікатор простору імен має бути URL, контрольованим прикладною схемою організації автора. Цільовий простір імен декларують у прикладній схемі з допомогою атрибуту targetNamespace елемента схеми з XML схеми.

Прикладна схема може охоплювати багато документів схем, що декларують той же самий простір імен.

**Примітка 1.** Рекомендовано, щоб документ схеми вищого рівня у такій диференційованій прикладній схемі прямо чи непрямо включав інші елементи для уникнення обмежень обробки при застосуванні XML, розглянутих у додатку М.

### 21.2.3 Імпорт GML схеми

GML прикладна схема має імпортувати повну GML схему. Вона може ідентифікувати GML профіль, що включає всі компоненти, які входять до GML, які вона прямо чи непрямо використовує для визначення її словника як визначено в підрозділі 20.5.

Необхідний імпорт компонентів GML схеми можна забезпечити через імпорт іншої схеми в просторі імен GML, що включає потрібні GML документи схеми.

*Приклад 1.* Імпорт gml.xsd з додатку В задовольнив би всі ці вимоги імпорту схеми.

```
<import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../gml.xsd"/>
```

Елемент <import> визначає, що компоненти, описані в документі імпортованої GML схеми, асоційовані з GML простором імен <http://www.opengis.net/gml/3.2>. Ідентифікатор цього простору імен має бути узгоджений із цільовим простором імен, визначеним у схемі, яку імпортують для забезпечення дійсності XML схеми.

Частка (schemaLocation) до імпортованого GML документу схеми має бути надана та може бути локальною копією документу, або URI посиланням на копію документу схеми в тому ж самому віддаленому сховищі.

*Приклад 2.* Приклади для таких сховищ подані на [http://www.iso.org/ittf/ISO\\_19136\\_Schemas](http://www.iso.org/ittf/ISO_19136_Schemas), на веб-сайті ISO або <http://schemas.opengis.net/> на веб-сайті OGC.

**Примітка 2.** Згідно із специфікацією W3C XML схеми, атрибут schemaLocation є тільки hint для фізичного розміщення та може бути пропущений аналізаторами XML.

#### **21.2.4 Отримання типу об'єкту**

Тип об'єкту, задекларований у прикладній схемі GML, не може порушувати будь-які обмеження на походження згідно до XML схеми, накладені атрибутом final на його базовий GML тип, та будь-які інші правила XML схеми.

Модель контенту типу об'єкту, визначеного у GML прикладній схемі, має походити з найбільш спеціалізованого GML типу об'єкту, що має виступати як основа для цієї моделі контенту, зберігаючи при цьому семантичну узгодженість та підвищуючи спеціалізацію типу.

#### **21.2.5 Елементи, якими подають об'єкти**

GML прикладна схема має декларувати глобальний елемент для будь-якого типу об'єкту, що виступає як кореневий елемент у GML документі.

#### **21.2.6 Отримання типу властивості**

Тип властивості, визначений для GML прикладної схеми для вмісту окремого GML об'єкту як включеного або посилання на нього, можна отримати від gml:AssociationRoleType, або він може відповідати шаблону цього типу.

Тип властивості, визначений у GML прикладній схемі для вмісту одного GML об'єкту, можна отримати від gml:InlinePropertyType або він може відповідати шаблону цього типу.

Тип властивості, визначений у GML прикладній схемі для посилання на один GML об'єкт, можна отримати від gml:ReferenceType, або він може відповідати шаблону цього типу.

Тип властивості, визначений прикладною схемою для вмісту однорідної колекції GML об'єктів, має відповідати шаблону gml:InlinePropertyType, але можна змінювати значення minOccurs та maxOccurs в елементі <sequence>.

**Примітка 1.** Оскільки отримання типів властивості через обмеження спричиняє проблеми з XML аналізаторами, загальноживаними в минулому, всі екземпляри таких отриманих типів видалені з GML схеми. Рекомендовано уникати отримання через походження в типах властивості в прикладних схемах.

#### **21.2.7 Елементи, якими подають властивості**

Елементи, якими подають властивості GML об'єктів, можна декларувати як глобальні елементи в прикладній схемі, або локально в моделях контенту об'єктів (визначень типу).

**Примітка 1.** Елементи в моделі контенту комплексних типів, визначені локальними іменами у прикладній схемі, відвертатимуть походження через обмеження в іншому просторі імен. Такі комплексні типи підходять для елементів, призначених для використання "як є" в їх власному просторі імен та можуть бути задекларовані як final="restriction". Елементи в контексті комплексних типів визначені шляхом посилань на глобальні елементи, що підтримують походження шляхом обмеження в іншому просторі імен, дозволяючи обмеження

потужності множини, та (або) заміну членом групи заміни. Такі комплексні типи розроблені для походження через обмеження, підходять у якості “бібліотечних типів” для елементів у групах заміни, які перетинають простори імен.

Якщо очікують, що значення властивості наявне десь іще, тип елементу властивості має підтримувати посилання на GML об’єкт, що є значенням властивості (див. 21.2.6).

Якщо очікують, що значення властивості подане шляхом включення, тип елементу властивості має підтримувати його або завдяки наявності простого контенту XML схеми відповідного простого типу, або шляхом включення GML об’єкту, що є включеним значенням властивості (див. пункт 21.2.6).

Якщо очікують, що значення властивості наявне десь іще або подане як включене, то тип елементу властивості має підтримувати обидва методи. У такому випадку тип елементу властивості повинен мати `gml:AssociationAttributeGroup`, де всі члени необов’язкові, а успадкований елемент повинен мати `minOccurs="0"`, щоб у документі екземпляру елемент властивості міг бути порожнім, якщо він містить `xlink` (див. пункт 7.2.3). Рекомендовано забороняти можливість одночасного використання чи невикористання атрибутів `xlink` та контенту, тоді це обмеження можна записати як нормативну директиву в елементі `<annotation>` на декларації елементу в GML прикладній схемі. Директива має бути виражена як проза або з використанням формальної нотації, наприклад `Schematron`.

### **21.3 Схеми, що визначають географічні об’єкти та колекції географічних об’єктів**

#### **21.3.1 Вступ**

Географічні об’єкти та їх колекції є первинною формою геопросторової інформації, що підтримується у GML, і вони зокрема корисні в моделюванні географії реального світу або у визначенні типів повідомлень для географічних веб-сервісів. Географічний об’єкт є засобом моделювання об’єкту реального світу або концепції (див. розділ 9).

Прикладні схеми географічних об’єктів визначають географічні об’єкти та їх колекції для спеціального прикладного домену або спільноти. Ці GML прикладні схеми повинні відповідати додатковим правилам, описаним у подальших підрозділах.

#### **21.3.2 Компоненти GML схеми імпорту**

Прикладна схема має імпортувати GML схему, як це описано в пункті 21.2.3. Будь-який GML профіль, на який посилається прикладна схема, має включати щонайменш елемент `gml:AbstractFeature` та всі компоненти схеми, використані для цим елементом.

#### **21.3.3 Елементи, якими подають об’єкти**

Всі географічні об’єкти та їх колекції у прикладній схемі треба декларувати як глобальні елементи у схемі, тобто вони мають бути безпосередньо успадкованими елементами елементу `<schema>` XML схеми.

Ім’я елементу, що надає екземпляри GML географічного об’єкту, має бути типом його географічного об’єкту в розумінні, описаному в ISO 19109.

Елемент, яким подають географічний об’єкт, має бути або безпосередньо, або непрямо в групі заміни, що включає `gml:AbstractFeature`.

Елемент, яким подають колекцію географічних об'єктів, має бути або безпосередньо, або непрямо в групі заміни для `gml:AbstractFeature` та мати в своїй моделі контенту елемент, модель контенту якого отримують шляхом розширення від `gml:AbstractFeatureMemberType`.

У GML прикладній схемі об'єкт, що є абстракцією реального світу, треба моделювати як географічний об'єкт. Всі інші об'єкти треба моделювати як GML об'єкт, що не є географічним, тобто елемент, яким подають об'єкт, має бути або безпосередньо, або непрямо в групі заміни, що включає `gml:AbstractGML`, але ніяким чином (безпосередньо чи непрямо) не в групі заміни, яка включає `gml:AbstractFeature`.

### 21.3.4 Географічні об'єкти аплікації є географічними об'єктами

Географічний об'єкт, визначений у прикладній схемі має відповідати правилам, що стосуються GML географічних об'єктів, як вони описані в розділі 9, та відповідати правилам, описаним у розділі 7.

**Примітка 1.** Ім'я елементу географічного об'єкту є семантичним типом цього об'єкту.

**Примітка 2.** Нащадки елементу географічного об'єкту є завжди елементами властивості, що описують цей об'єкт, та такі властивості завжди закодовані як успадковані елементи. Властивості не кодують як XML атрибути.

### 21.4 Схеми, що визначають просторові геометрії

#### 21.4.1 Компоненти імпорту GML схеми геометрії

Прикладна схема має імпортувати GML схему, як описано в пункт 21.2.3. Будь-який GML профіль, на який посилається прикладна схема, включає щонайменш елемент `gml:AbstractGeometry` та всі компоненти схеми, використані цим елементом.

**Примітка 1.** Звичайно додаткові компоненти схеми геометрії потрібні в додаток до тих, що необхідні для `gml:AbstractGeometry`. На практиці зокрема конкретні елементи, такі як `gml:Point` та `gml:PointPropertyType` звичайно є частинами профілю.

#### 21.4.2 Типи геометрії та типи властивостей геометрії, визначені користувачем

##### 22.4.2.1 Типи геометрії, визначені користувачем

Автори прикладних схем можуть створювати свої власні типи геометрії, якщо у GML бракує бажаних конструктивів. Щоб робити це, автори мають переконатися, що елементи об'єкту цих конкретних типів геометрії та колекції геометрій знаходяться в групі заміни (або прямо, або непрямо) відповідного елементу `gml:AbstractGeometry` GML об'єкту.

*Приклад 1.* Наступне визначення типу елементу та комплексу в прикладній схемі розширює `gml:Point` та додає напрямок (тобто орієнтацію символу у зображенні).

```
<element name="PointWithBearing" type="ex:PointWithBearingType"
substituionGroup="gml:Point">

  <complexType name="PointWithBearingType">
    <complexContent>
      <extension base="gml:PointType">
        <sequence>
          <element name="bearing" type="gml:AngleType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

Будь-які визначені користувачем підтипи геометрії мають успадковувати елементи та атрибути цих базових GML типів геометрії без обмеження, але можуть розширювати ці базові типи для задоволення вимог аплікації, таких як краща ступінь інтероперабельності із спадкоємними системами та наборами даних.

Всі правила, визначені в розділах 7, 10, пункт 10.5.10 та підрозділі 11, треба виконувати.

#### **21.4.2.2 Типи властивості геометрії, визначені користувачем**

Більше того, автори прикладних схем можуть створювати власні типи властивості геометрії, що охоплюють геометричні типи, визначені в розділі 10, пункт 10.5.10 чи в розділі 11, або визначені згідно з підпунктом 21.4.2.1. Вони надають гарантію, що ці властивості відповідають шаблону, використаному в `gml:GeometryPropertyType` для стандартних властивостей та `gml:GeometryArrayPropertyType` для властивостей масиву. Цільовим типом має бути *bona fide* елементу об'єкту геометрії.

Тип властивості геометрії може бути обмеженням на `gml:GeometryPropertyType`, але це не є вимогою. Тим не менше, кожна властивість геометрії має відповідати шаблону цього типу. Прикладна схема має підтримувати вибір між семантиками включення та посилання або може обмежувати вживання або включеннями (забороняти використання атрибутів `Xlink`), або посиланнями (забороняти включення геометрії в географічний об'єкт).

Тип властивості масиву геометрії може бути обмеженням на `gml:GeometryArrayPropertyType`, але це не є вимогою. Тим не менше, кожна властивість геометрії має відповідати шаблону цього типу. Всі елементи геометрії у масиві містяться як включені в об'єкті, і тільки семантики включення підтримуються властивостями масиву.

*Приклад 1.* Наступні визначення комплексного типу в прикладній схемі визначають "стандартний" тип властивості для типу геометрії та типу властивості масиву, визначених користувачем для такого ж типу геометрії.

```
<complexType name="MyGeometryPropertyType">
  <sequence>
    <element ref="foo:PointWithBearingType" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup" />
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

<complexType name="MyGeometryArrayPropertyType">
  <sequence>
    <element ref="foo:PointWithBearingType" minOccurs="0" maxOccurs="unbounded" />
  </sequence>
</complexType>
```

### **21.5 Схеми, що визначають просторові топології**

#### **22.5.1 Компоненти GML схеми імпорту топології**

Прикладна схема має імпортувати необхідні компоненти з GML, як це описано в пункті 21.2.3. Будь-який профіль, на який спрямоване посилання від прикладної схеми,

має включати щонайменш елемент `gml:AbstractTopology` та всі компоненти схеми, використані цим елементом.

**Примітка 1.** Звичайно додаткові компоненти схеми топології потрібні у додаток до тих, що необхідні для `gml:AbstractTopology`. Зокрема конкретні елементи та типи, подібні до `gml:Edge` та `gml:DirectedEdgePropertyType`, на практиці звичайно треба імпортувати.

*Приклад 1.* GML схема імпорту (іде далі) з використанням документи схеми додатку B:

```
<import namespace="http://www.opengis.net/gml" schemaLocation="../topology.xsd"/>
```

## **21.5.2 Визначені користувачем типи топології та властивостей топології**

### **21.5.2.1 Визначені користувачем типи топології**

Автори прикладних схем можуть створювати свої власні типи топології, якщо у GML бракує бажаних конструктив. При цьому автори мають переконатися, що елементи об'єкту цих конкретних типів топології знаходяться в групі заміни (або прямо, або непрямо) відповідного елементу GML об'єкту: `gml:AbstractTopology`.

Будь-які підтипи визначеної користувачем топології мають успадковувати елементи та атрибути базових GML типів топології без обмежень, але можуть розширювати ці типи для задоволення вимог аплікації, таких як забезпечення кращої інтероперабельності із спадкоємними системами та наборами даних.

Треба виконувати всі правила, визначені в розділах 7 та 13.

### **21.5.2.2 Визначені користувачем типи властивостей топології**

Крім цього, автори прикладних схем можуть створювати свої власні (спрямовані) типи властивостей топології, які інкапсулюють типи топології, визначені згідно до розділу 13. Вони мають переконатися в тому, що ці властивості відповідають правилам, описаним у пункті 21.2.6. У додаток, цільовий тип має бути надійним топологічним конструктом.

Тип властивості топології може бути обмеженням існуючого типу властивості топології.

Тип властивості топології може підтримувати вибір між семантикою включення та посилань, або може обмежувати використання або включень (забороняти використання атрибутів `Xlink`), або посилань (забороняти вміст геометрії в географічному об'єкті).

## **21.6 Схеми, що визначають час**

### **21.6.1 Компоненти GML часової схеми імпорту**

Прикладна схема має імпортувати необхідні компоненти з GML, як описано в пункті 21.2.3. Будь-який профіль, на який спрямоване посилання від прикладної схеми, має включати щонайменше елемент `gml:AbstractTimeObject` та всі компоненти схеми, які використовують для цього елементу.

**Примітка 1.** Звичайно додаткові компоненти часової схеми потрібні у додаток до тих, які вимагає `gml:AbstractTimeObject`. На практиці зокрема конкретні елементи та типи, подібні до `gml:TimeInstant` та `gml:TimeInstantPropertyType` будуть частиною профілю.

### **21.6.2 Часові типи та типи часової властивості, визначені користувачем**

#### **21.6.2.1 Часові типи, визначені користувачем**

Автори прикладних схем можуть створювати свої власні часові типи, якщо у GML бракує бажаних засобів побудови. Роблячи це, автори мають переконатися в тому, що елементи об'єктів цих конкретних часових типів знаходяться в групі заміни (або прямо, або непрямо) відповідного елементу GML об'єкту: `gml:AbstractTimeObject`.



Будь-який часовий підтип, визначений користувачем, має успадковувати елементи та атрибути базових часових GML типів без обмежень, але можуть розширювати ці базові типи для задоволення прикладних вимог, таких як забезпечення кращої інтероперабельності із спадкоємними системами та наборами даних.

Всі правила, визначені в розділах 7 та 14, треба виконувати.

#### **21.6.2.2 Типи часових властивостей, визначені користувачем**

Крім того, автори прикладних схем можуть створювати свої власні типи часових властивостей, що інкапсулюють часові типи, які визначають згідно з розділом 14. Вони мають переконатися, що ці властивості відповідають правилам, описаним у пункті 21.2.6. У додаток, цільовий тип має бути надійним часовим конструктом.

Тип часової властивості може бути обмеженням існуючого типу часової властивості.

Тип часової властивості може підтримувати вибір між семантиками включення та посилання або може обмежувати використання до або включення (не можна використовувати атрибути Xlink), або посилання (не можна містити геометрію в географічному об'єкті).

### **21.7 Схеми, що визначають референцні системи координат**

#### **21.7.1 Вступ**

Багато конкретних XML елементів, визначених у схемах референцних систем координат, можна використовувати без прикладних схем, завжди, коли не потрібно жодних розширень або обмежень. Прикладну схему треба використовувати завжди, коли потрібне розширення контенту елемента, а також у багатьох інших випадках для визначення потрібних обмежень. Тобто, прикладна схема повинна бути визначена для розширення та (або) обмеження елементів для специфічної аплікації чи набору аплікацій з метою:

- додавання елементів до контентів існуючих елементів для запису додаткових даних про деякий аспект, потрібних для даної аплікації;
- обмежування множинності поточних елементів контентів, для усунення гнучкості, яка може й не бути потрібною та, можливо, ускладнює дану аплікацію;
- використання відмінного імені елемента для полегшення розуміння в певній аплікації, головне для елементів, що можуть неодноразово проявлятися в екземплярах;
- визначення стандартних контентів та шаблони контентів для вибраних елементів та атрибутів, що потрібно для інтероперабельності.
- визначення стандартних XML та інших документів, на які будуть спрямовані посилання чи які будуть використовувати в інший спосіб для покращення інтероперабельності.

Прикладні схеми можна таким чином використати для розширень чи (та) обмежень контентів XML документів. Обмеження контенту очікують як часто використовуване для того, щоб обмежувати контент з метою підвищення інтероперабельності та уникнення неоднозначностей, коли значна гнучкість не потрібна в аплікації. Розширення існуючих конкретних елементів можна визначити шляхом розширення цього конкретного елемента. Часто обмеження існуючих конкретних елементів можна виконати, розширивши абстрактний елемент, з якого походить цей конкретний елемент, шляхом додавання якихось інших, але відповідних розширень.

Прикладна схема може визначати єдиний елемент найвищого рівня для використання в XML документі з XML елементами та типами, що використовують у ньому. Цей єдиний XML елемент найвищого рівня може бути об'єктом з ідентифікацією, але це не обов'язково. Така прикладна схема буде імпортувати та будувати на XML компонентах схеми, визначених у розділі 12.

Прикладні схеми можуть визначати додаткові конкретні елементи з використанням чи розширенням інших абстрактних елементів, якщо це потрібно. Однак, додатковий конкретний елемент, що використовує чи розширює абстрактний елемент, не слід визначати, якщо цей конкретний елемент суттєво подібний до існуючого елемента, і тому, можливо, треба розширяти чи використовувати існуючий конкретний елемент. Часто існуючі конкретні елементи, в яких використовують абстрактний елемент, вважають значною мірою вичерпними. Це зокрема вірно, коли існуючі конкретні елементи включають один елемент, що є достатньо загальним, як, наприклад, елементи EngineeringCRS, DerivedCRS, EngineeringDatum, UserDefinedCS, OperationParameter та OperationParameterGroup.

Елементи Conversion, Transformation, ParameterValue, та ParameterValueGroup можна використовувати для добре відомих методів координатних операцій, особливо коли тільки один екземпляр елемента потрібний для цього методу операції. Однак, ці елементи не слід використовувати для добре відомих методів координатних операцій, якщо багато екземплярів цього елемента потрібні для одного методу операції. Натомість, треба підготувати прикладну схему, яка визначає імена та контенти елементів спеціально для методу операції та контенти, для кожного такого методу операції. Для інтероперабельності відповідна геоінформаційна спільнота має стандартизувати кожен таку прикладну схему.

**Примітка 1.** Використання прикладних схем відповідає шаблону, використаному в прикладних схемах географічних об'єктів.

### 21.7.2 Імпорт компонентів GML схеми референційної системи координат

Прикладна схема має імпортувати GML схему, як описано в пункті 21.2.3. Це включає щонайменш елемент gml:AbstractCoordinateReferenceSystem та всі компоненти схеми, використані в цьому елементі.

**Примітка 1.** Звичайно додаткові компоненти схеми референційної системи координат потрібні в додаток до тих, які вимагає gml:AbstractCoordinateReferenceSystem. На практиці, особливо конкретні елементи та типи, звичайно також будуть частинами цього профілю.

## 21.8 Схеми, що визначають покриття

### 21.8.1 Вступ

У наступних підрозділах визначені правила для побудови прикладних GML схем для покриттів. Покриття визначені в розділі 19. Треба зауважити, що покриття – це географічні об'єкти, і тому правила з підрозділу 21.3 треба застосовувати і для покриттів.

### 21.8.2 Імпорт компоненти GML схеми покриття

Прикладна схема має імпортувати GML схему, як це описано в пункті 21.2.3. Будь-який профіль GML включає щонайменш елемент gml:AbstractCoverage та всі компоненти схеми, використані в цьому елементі.

**Примітка 1.** Треба зауважити, що додаткові компоненти схеми покриття звичайно потрібні в додаток до тих, які вимагає gml:AbstractCoverage. Зокрема, на практиці конкретні

елементи та типи, подібні до `gml:RectifiedGridCoverage`, також, як правило, будуть частиною профілю.

### 21.8.3 Типи покриттів, визначені користувачем

Всі географічні покриття в GML прикладній схемі треба декларувати як глобальні елементи у схемі, тобто вони мають бути успадкованими елементами елементу `<schema>` XML схеми.

Автори прикладних схем можуть створювати свої власні типи покриттів, якщо GML бракує потрібних конструктивів. Для цього автори мають переконатися, що елементи об'єкту цих типів покриття присутні у групі заміни (або прямо, або непрямо) відповідного GML елементу об'єкту: або `gml:AbstractDiscreteCoverage`, або `gml:AbstractContinuousCoverage`.

**Примітка 1.** Ці елементи присутні непрямо чинном у групі заміни `gml:AbstractFeature`, тому умова моделі географічного об'єкту дотримана.

**Примітка 2.** Цей міжнародний стандарт забезпечує конкретні типи покриття `gml:MultiPointCoverage`, `gml:MultiCurveCoverage`, `gml:MultiSurfaceCoverage`, `gml:MultiSurfaceCoverage`, `gml:GriddedCoverage`, та `gml:RectifiedGridCoverage`. Прикладні покриття можна отримувати з будь-якого з перелічених типів.

Будь-який визначений користувачем підтип покриття має успадковувати елементи та атрибути цих базових GML типів покриття без обмеження, але й може також розширювати ці базові типи для виконання вимог аплікації, таких, як забезпечення кращої інтероперабельності із спадкоємними системами та наборами даних.

Всі правила, визначені в розділах 7, 9 та 19, треба виконувати.

### 21.8.4 Параметри діапазону мають бути здатними замінювати `AbstractValue`

Прикладна схема покриття має визначати або імпортувати визначення для всіх параметрів діапазону. Кожний такий параметр має бути таким, що може замінити `gml:AbstractValue`, як визначено в підрозділі 16.4. Треба зауважити, що це дозволяє різні типи параметру діапазону. Крім того, компоненти схеми, визначені в підрозділі 16.4, включають кілька абстрактних підтипів, які можуть замінювати `gml:AbstractValue`, включаючи `gml:AbstractScalarValue` та `gml:AbstractValueList`. Конкретні типи списків значень та скалярів та головні елементи груп заміни також подані (якими можна замінити `gml:AbstractScalarValue` або `gml:AbstractValueList`) та включають:

- `gml:Category` (модель контенту визначена з допомогою `gml:CodeType`);
- `gml:CategoryList` (модель контенту визначена з допомогою `gml:CodeOrNilReasonListType`);
- `gml:Quantity` (модель контенту визначена з допомогою `gml:MeasureType`);
- `gml:QuantityList` (модель контенту визначена з допомогою `gml:MeasureOrNilReasonListType`);
- `gml:Count` (модель контенту визначена з допомогою `gml:CountType`);
- `gml:Boolean` (модель контенту визначена з допомогою `gml:BooleanType`).

Для визначення діапазону параметрів у прикладній схемі покриття див. підрозділ 16.4.

*Приклад 1.* Типові приклади використання типів значень у розробці GML покриття можна знайти в підрозділі 19.3; вони підсумовані в таблиці 5.

**Таблиця 8** – Параметри діапазонів для схем покриття

<b>Покриття</b>	<b>Параметр діапазону</b>	<b>Визначення у GML</b>
Розподіл температур (погода)	Температура	Була б отримана з <code>gml:MeasureOrNilReasonListType</code> та подана такою, що може замінити <code>gml:measure</code> , визначений у 16.3.2.
Розподіл типів ґрунтів (агрономія)	Тип ґрунтів	Був би отриманий з <code>gml:CategoryType</code> та поданий таким, що може замінити <code>gml:Category</code> . Слабке посилання на перелік типів ґрунтів.
Багатоспектральне оптичне зображення (дистанційне зондування)	Відбиття у кожному спектральному каналі	Було б отримане з <code>gml:QuantityListType</code> та подане таким, що може замінити <code>gml:QuantityList</code> .
Розподіл випадків вірусу Західного Нілу (епідеміологія)	CaseCount (рахунок випадків)	Був би отриманий з <code>gml:integerOrNilReasonList</code> , та поданий таким, що може замінити <code>gml:CountList</code> .

### 21.8.5 Документ покриття

Документ покриття визначений відповідною схемою покриття. Кореневий елемент цього документу має бути покриттям, визначений у цій схемі або може бути колекцією географічних елементів, як описано в підрозділі 9.9, члени якої є покриттями.

### 21.9 Схеми, що визначають спостереження

#### 21.9.1 Вступ

Наступні підкласи описують, як створювати прикладну схему спостережень. Спостереження визначені в розділі 18. Прикладна схема спостережень визначає один чи більше типів спостережень згідно до нижчевикладених правил.

#### 21.9.2 Імпорт компонентів GML схеми спостереження

Прикладна схема має імпортувати GML схему, як описано в пункті 21.2.3. Будь-який профіль GML, на який спрямоване посилання від прикладної схеми, включає щонайменше елемент `gml:Observation` та всі компоненти схеми, використані цим елементом.

#### 21.9.3 Типи спостережень, визначені користувачем

Всі типи спостереження, визначені в прикладній GML схемі, треба декларувати як глобальні елементи в схемі, тобто вони мають бути успадкованими елементами елементу `<schema>` XML схеми. Модель контенту для таких глобальних елементів треба отримувати шляхом розширення або прямо, або непрямо від `gml:ObservationType`.

Автори прикладних схем можуть створювати свої власні типи спостережень, якщо GML бракує потрібних конструктів. Для цього автори мають переконатися, що елементи об'єктів цих спостережень цих типів спостережень є в групі заміни (або прямо, або непрямо) відповідного GML елементу об'єкту: `gml:Observation`.

**Примітка 1.** Ці елементи непрямо присутні в групі заміни `gml:AbstractFeature`, і тому умова моделі географічного об'єкту дотримана.

**Примітка 2.** Цей міжнародний стандарт надає конкретні типи `gml:Observation`, `gml:DirectedObservation` та `gml:DirectedObservationAtDistance`. Прикладні типи спостереження можна отримувати з будь-яких перелічених вище.

Всі правила, визначені у розділах 7, 9 та 18, треба виконувати.

#### **21.9.4 Колекції спостережень**

Всі колекції спостережень у GML прикладній схемі треба декларувати як глобальні елементи в схемі, тобто вони мають бути успадкованими елементами елементу `<schema>` XML схеми. Колекція спостережень має бути колекцією географічних об'єктів, як це описано в підрозділі 9.9, члени якої є спостереженнями.

#### **21.9.5 Спостереження як географічні об'єкти**

Спостереження, визначені в прикладній схемі, мають відповідати правилам стосовно GML географічних об'єктів, як описано в розділах 7 та 9. Див. також п.пункт 21.3.4.

#### **21.9.6 Документ колекції спостережень**

Згідно до правил для GML документів (див. підрозділ 21.1), документ колекції спостережень може посилатися на спостереження, що визначені в будь-якому числі GML прикладних схем, і вони можуть визначати тільки спостереження, колекції спостережень або будь-яку комбінацію першого і другого.

#### **21.10 Схеми, що визначають словники**

##### **21.10.1 Вступ**

У наступних пунктах описано, як створювати прикладну схему для визначень. Визначення та словники описані в розділі 15. Один набір спеціалізованих визначень, вбудований у GML – для одиниць вимірювання, та слугує як приклад отримання компонентів визначення.

Прикладна схема для визначень визначає один чи більше типів визначень згідно до наступних правил.

##### **21.10.2 Компоненти імпорту словника GML схеми**

Прикладна схема має імпортувати GML схему, як описано в пункті 21.2.3. Будь-який профіль GML включає щонайменш елемент `gml:Definition` та всі компоненти схеми, використані цим елементом.

##### **21.10.3 Типи визначень, визначені користувачем**

Всі визначення в прикладній схемі треба декларувати як глобальні елементи в схемі, тобто вони мають бути безпосередньо успадкованими елементами елементу `<schema>` XML схеми. Модель контенту для таких глобальних елементів треба отримувати або прямо, або непрямо від `gml:DefinitionType`.

Автори прикладних схем можуть створювати свої власні типи визначень, якщо GML бракує потрібних конструктів. Для цього автори мають переконатися, що елементи об'єкту цих типів визначення є в групі заміни (або прямо, або непрямо) відповідного елементу GML об'єкту: `gml:Definition`.

Всі правила, визначені в розділах 7 та 15, треба виконувати.

##### **21.10.4 Визначені користувачем типи словників**

Словник у прикладній схемі треба декларувати як глобальний елемент у схемі, тобто він має бути успадкованим елементом елементу `<schema>` XML схеми. Модель контенту для таких глобальних елементів треба отримувати або прямо, або непрямо від `gml:DictionaryType`.

Автори прикладних схем можуть створювати свої власні типи словників, якщо GML бракує потрібних конструктивів. Для цього автори мають переконатися, що елементи об'єкту цих типів визначення є в групі заміни (або прямо, або непрямо) відповідного елементу GML об'єкту: `gml:Dictionary`.

Всі правила, визначені в розділах 7 та 16, треба виконувати.

## **21.11 Схеми, що визначають значення**

### **21.11.1 Вступ**

GML дозволяє користувачеві визначати типи значень. Типи значень можна використовувати для того, щоб виразити типи властивостей географічних об'єктів та інші типи GML об'єктів. Базові кореневі типи для значень, визначених користувачем, визначені в підпункті 7.2.2.1. Альтернативна форма для подання значень міститься в підрозділі 16.4. Це використовують головне для надання значень параметру `gml:resultOf` для спостережень.

### **21.11.2 Компоненти імпорту GML схеми об'єктів значень**

Прикладна схема має GML схему, як описано в пункті 21.2.3. Будь який профіль GML, на який спрямоване посилання від прикладної схеми, має включати щонайменше групу `gml:Value` та всі компоненти схеми, використані цим елементом.

### **21.11.3 Побудова нових типів значень**

Нові значення можна створювати шляхом отримання (звичайно шляхом обмеження) з будь-якого з кореневих типів, показаних у таблиці 6.

**Таблиця 9** – Побудова нових типів

Модель контенту	Опис
<code>gml:MeasureType</code>	Числова величина з одиницею вимірювання ( <code> uom</code> )
<code>gml:CategoryType</code>	Класифікація
<code>gml:CountType</code>	Рахунок випадків, інцидентів і т. і.

Деякі стандартні типи значення можна знайти в компонентах схеми, визначених у підрозділі 16.3.

## **21.12 GML профілі GML схеми**

Звичайно GML прикладна схема вимагатиме тільки обмежений піднабір компонентів схеми GML схеми. Рекомендовано ідентифікувати та документувати GML профіль (див. розділ 20), який вимагає GML прикладна схема. Підрозділи з 21.3 по 21.11 містять деякі інструкції, які компоненти схеми мають бути потрібними залежно від типу GML прикладної схеми.

**Примітка 1.** Додаток Ж містить метод автоматичного створення GML профілю, базованого на списку компонентів схеми, яку експліцитно вимагає GML прикладна схема.

Передусім треба враховувати такі інструкції:

- у прикладній схемі, в якій моделюють географічні об'єкти, мають бути `gml:AbstractFeature` та всі компоненти схеми, яких вимагає цей елемент (див. розділ 9);
- у прикладній схемі, в якій моделюють також колекції географічних об'єктів, мають бути також `gml:AbstractFeatureMemberType` та `gml:AggregationAttributeGroup`, як і всі компоненти схеми, яких вони вимагають (див. розділ 7);
- якщо географічні об'єкти мають властивості, в яких використовують одиниці вимірювання, то необхідні `gml:MeasureType` та всі спеціальні підтипи, тобто `gml:LengthType`, як і всі компоненти схеми, яких вони вимагають (див. розділ 8).

- `gml:BaseUnit`, `gml:DerivedUnit` та (або) `gml:ConventionalUnit` (та всі компоненти схеми, яких вони вимагають, див. розділ 7) не потрібні, доки аплікація не визначає одиниці вимірювання так, як це визначення зустрічається у словнику одиниць вимірювання;
- якщо в прикладній схемі використовують тільки нульвимірні просторові геометрії, виникає необхідність у `gml:Point` (та у всіх компонентах схеми, яких він вимагає), див. розділ 10;
  - якщо в прикладній схемі використовують тільки одновимірні просторові геометрії з лінійною інтерполяцією, потрібен тільки `gml:LineString` (та у всіх компонентах схеми, яких він вимагає), див. розділ 10;
  - якщо в прикладній схемі використовують додаткові типи інтерполяції, потрібні `gml:Curve` та будь-яке число сегментів кривої залежно від аплікації (але хоча б один); знову, це включає всі компоненти схеми, які потрібні для цих елементів (див. пункт 10.5.10);
  - якщо в прикладній схемі використовують тільки двохвимірні просторові геометрії з лінійною інтерполяцією вздовж їх меж без спільних елементів межі, потрібні тільки `gml:Polygon` та `gml:LinearRing` (та всі компоненти схеми, яких вони вимагають), див. розділ 10;
  - якщо в прикладній схемі використовують додаткові типи інтерполяції або ділянки поверхонь, потрібні `gml:Surface` та будь-яке число ділянок поверхні залежно від аплікації (але хоча б одна); якщо поверхні повинні мати спільні геометричні примітиви вздовж їх меж, то також потрібний `gml:Ring`; знову, це включає всі компоненти схеми, які потрібні для цих елементів (див. пункт 10.5.10);
  - якщо в прикладній схемі використовують трьохвимірні просторові геометрії, потрібний `gml:Solid` (та всі компоненти схеми, яких він вимагає), див. пункт 10.5.10;
  - схема компонентів геометричних агрегатів, описана в розділі 11, потрібна, тільки якщо географічні об'єкти використовують геометричні об'єкти, що є колекціями геометричних примітивів у своїх просторових властивостях;
  - геометричний комплекс та схема композитів, описані в розділі 11, потрібні, тільки якщо географічні об'єкти використовують геометричні комплекси в їх просторових властивостях;
  - компоненти схеми топології, описані в розділі 13, потрібні, тільки якщо географічні об'єкти мають властивості топології;
  - компоненти схеми референцної системи координат, описані в розділі 12, потрібні, тільки якщо аплікація вимагає побудови або обробки записів у словнику референцних систем координат (та компонентів, що підтримують їх, таких, як, наприклад, меридіани, геодезичні дати);
  - часові схеми, описані в розділі 14, потрібні, тільки якщо прикладна схема має враховувати властивості географічних об'єктів, залежних від часу, або динамічні об'єкти;
  - схема компонентів покриттів, описана в розділі 19, потрібна, тільки якщо прикладна схема передбачає побудову чи обробку покриттів (наприклад, зображення, отримані шляхом дистанційного зондування, аерофотознімки, розподіл ґрунтів, моделі рельєфу є типовими покриттями);

- компоненти схеми спостережень, описані в розділі 18, потрібні, тільки якщо у прикладній схемі треба врахувати моделювання актів спостережень, такі, як отримання фотографій або вимірювання; у другому випадку об'єкти значень та компоненти схеми вимірів також звичайно потребуються;
- компоненти схеми напрямку, описані в розділі 17 потрібні, тільки якщо прикладна схема вимагає засобів побудови напрямків, таких як компасний пеленг; схему напрямку використовують у `gml:DirectedObservation`.
- елементи властивості, модель контенту яких отримана шляхом розширення з `gml:AbstractMetadataPropertyType` (див. пункт 7.2.6), використовують для визначення прикладних схем метаданих для специфічних об'єктів.

**Примітка 1.** У багатьох аплікаціях треба тільки імпортувати `feature.xsd` of Annex C, оскільки це транзитивно імпортує схеми геометрії та `gmlBase.xsd`. Для докладного розгляду залежностей схеми та модульності див. Додаток Л.

## ДОДАТОК А (нормативний)

### АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ GML ПРИКЛАДНИХ СХЕМИ, GML ПРОФІЛІВ ТА GML ДОКУМЕНТІВ

#### **A.1 Абстрактні комплекти тестів для GML прикладних схем**

##### **A.1.1 Випадки тестів для обов'язкових класів відповідності**

###### **A.1.1.1 Використання XML просторів імен**

1) Мета тесту: перевірити правильність використання XML просторів імен у GML прикладних схемах.

2) Метод перевірки: переконатися, що всі компоненти схеми в прикладній схемі асоційовані з XML простором імен та цей простір імен не є "<http://www.opengis.net/gml/3.2>".

3) Посилання: ISO 19136:2007, пункт 21.2.2.

4) Тип тесту: базовий тест.

###### **A.1.1.2 Загальні правила**

1) Мета тесту: переконатися, що GML прикладна схема відповідає загальним правилам побудови GML прикладних схем.

2) Метод тестування: перевірити прикладну схему та переконатися, що вона задовольняє загальні правила, описані в пункті 21.2.1.

4) Посилання: ISO 19136:2007, пункт 21.2.1.

д) Тип тесту: тест на спроможність.

###### **A.1.1.3 Імпорт компонентів GML схеми**

1) Мета тесту: переконатися, що GML прикладна схема імпортує повну GML схему та правильно посилається на GML профіль.

2) Метод тесту: перевірити положення імпорту у прикладній схемі (повна прикладна GML схема має бути прямо чи непрямо імпортована); крім того, якщо спрямовані посилання на один чи більше GML профілів, переконатися, що компоненти XML схеми, визначені в елементах `gml:gmlProfileSchema`, задовольняють усі обов'язкові вимоги до відповідності абстрактному комплекту тестів в A.2.1).

3) Посилання: ISO 19136:2007, підрозділ 20.5, пункт 21.2.3.



4) Тип тесту: тест на спроможність.

#### **A.1.1.4 Дійсна XML схема**

1) Мета тесту: перевірити дійсність XML документу GML прикладної схеми за специфікацією XML схеми.

2) Метод тесту: перевірити XML документ GML прикладної схеми за специфікацією XML схеми; процес може передбачати використання відповідного програмного інструменту перевірки або ручний процес перевірки всіх відповідних визначень із специфікації XML схеми.

3) Посилання: ISO 19136:2007, пункт 21.2; W3C XML Schema Part 1, W3C XML Schema Part 2 (див. розділ 3).

4) Тип тесту: тест на спроможність.

#### **A.1.1.5 Підтримка моделі та синтаксису GML**

1) Мета тесту: переконатися, що GML прикладна схема відповідає правилам кодування об'єктів та властивостей.

2) Метод тесту: перевірка прикладної схеми.

3) Посилання: ISO 19136:2007, підрозділ 7.1, розділ 21.

4) Тип тесту: тест на спроможність.

#### **A.1.1.6 Група заміни елементів об'єкту, отримання типу**

1) Мета тесту: переконатися, що всі об'єкти в GML прикладній схемі перебувають у коректній групі заміни;

2) Метод тесту: перевірити прикладну схему на входження всіх об'єктів із ідентичністю (пряме чи непряме) до групи заміни gml:AbstractGML; перевірити, що дотримано правил отримання з цих базових типів, викладених у A.1.1.6 (3);

3) Посилання: ISO 19136:2007 підрозділ. 6.2, пункти 7.2.2, 21.2.4, 21.2.5;

4) Тип тесту: тест на здатність.

#### **A.1.1.7 Елементи властивості, що не є елементами об'єкту**

1) Мета тесту: перевірити, що всі елементи властивості в прикладній схемі не є об'єктами.

2) Метод тесту: перевірити, що кожен успадкований елемент кожного елементу об'єкту не є прямо чи непрямо в групі заміни gml:AbstractObject.

3) Посилання: ISO 19136:2007 пункти 7.2.2, 21.2.5.

4) Тип тесту: тест на спроможність.

#### **A.1.1.8 Модель контенту елементів властивості**

1) Мета тесту: переконатися, що всі елементи властивості в GML прикладній схемі мають дійсну модель контенту.

2) Метод тесту: перевірити кожен успадкований елемент кожного елементу об'єкту в прикладній схемі.

3) Посилання: ISO 19136, пункти 7.2.3, 22.2.6, 21.2.7.

4) Тип тесту: тест на спроможність.

#### **A.1.1.9 Властивості метаданих та якості даних**

1) Мета тесту: переконатися, що всі властивості, значенням яких є метадані про об'єкт, можна ідентифікувати як властивість метаданих.

2) Метод тесту: перевірити GML прикладну схему на отримання моделі контенту всіх елементів властивості, значення яких є метаданими, шляхом розширення від `gml:AbstractMetadataPropertyType`.

3) Посилання: ISO 19136:2007, підпункт 7.2.7.1.

4) Тип тесту: тест на спроможність.

#### **A.1.1.10 Властивості просторової геометрії**

1) Мета тесту: переконатися, що всі властивості, значенням яких є об'єкт просторової геометрії, можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що всі властивості з геометричним об'єктом або колекцією таких об'єктів задекларовані відповідно до розділу 9.5.

3) Посилання: ISO 19136:2007, розділ 9.5.

4) Тип тесту: тест на спроможність.

#### **A.1.1.11 Властивості просторової топології**

1) Мета тесту: перевірити, що всі властивості, значенням яких є об'єкт просторової топології, можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що всі властивості з топологічним об'єктом або колекцією таких об'єктів зазначені відповідно до підрозділу 9.6.

3) Посилання: ISO 19136:2007, підрозділ 9.6.

4) Тип тесту: тест на спроможність.

#### **A.1.1.12 Часові властивості:**

1) Мета тесту: переконатися, що всі властивості, значенням яких є часовий об'єкт, можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що всі властивості з часовим об'єктом чи колекцією таких об'єктів задекларовані відповідно до підрозділу 9.7;

3) Посилання: ISO 19136:2007, підрозділ 9.7.

4) Тип тесту: тест на спроможність.

#### **A.1.1.13 Властивості місцеположення:**

1) Мета тесту: переконатися, що всі властивості із значенням опису місцеположення або посиланням можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що у всіх властивостях з просторовими посиланнями через географічні ідентифікатори використані елементи властивості `gml:locationName` або `gml:locationReference`.

3) Посилання: ISO 19136:2007, підпункт 9.1.4.3.

4) Тип тесту: тест на спроможність.

#### **A.1.1.14 GML колекції об'єктів**

1) Мета тесту: переконатися, що всі об'єкти, що є GML колекціями об'єктів, можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що такі об'єкти мають один чи більше елементів властивості з моделлю контенту, яка розширює `gml:AbstractMemberType`, а також, якщо доречно, що атрибут `gml:aggregationType` є успадкованим вузлом елементу об'єкту.

3) Посилання: ISO 19136:2007, пункт 7.2.5.

4) Тип тесту: тест на спроможність.

#### **A.1.1.15 Група заміни елементів географічного об'єкту**

1) Мета тесту: переконатися, що всі об'єкти в GML прикладній схемі перебувають у коректній групі заміни.

2) Метод тесту: перевірити прикладну схему на те, що всі елементи об'єктів, яким подано географічні об'єкти, прямо чи непрямо перебувають у групі заміни географічного об'єкту `gml:AbstractFeature`.

3) Посилання: ISO 19136:2007, підрозділ 9.3.

4) Тип тесту: тест на спроможність.

#### **A.1.1.16 GML колекції географічних об'єктів**

1) Мета тесту: переконатися, що всі географічні об'єкти, що є колекціями GML географічних об'єктів, можна ідентифікувати як такі.

2) Метод тесту: перевірити GML прикладну схему на те, що такі географічні об'єкти мають один чи більше елементів властивості з моделлю контенту, яка розширює `gml:AbstractFeatureMemberType`, а також, якщо доречно, що атрибут `gml:aggregationType` є успадкованим вузлом елементу об'єкту.

3) Посилання: ISO 19136:2007, підрозділ 9.9.

4) Тип тесту: тест на спроможність.

### **A.1.2 Випадки тестів GML прикладних схем, конвертованих з прикладних схем стандарту ISO 19109 в UML**

#### **A.1.2.1 Дійсні прикладні схеми стандарту ISO 19109**

1) Мета тесту: якщо GML прикладна схема отримана з прикладної схеми стандарту ISO 19109 в UML, то переконатися, що UML прикладна схема відповідає вимогам стандарту ISO 19109.

2) метод тесту: перевірити відповідність UML прикладної схеми стандарту ISO 19109 та перевірити, чи була UML прикладна схема побудована згідно до Д.2.1.

3) Посилання: ISO 19109, розділ 2, додаток А; ISO 19136:2007, Д.2.1.

4) Тип тесту: тест на спроможність.

#### **A.1.2.2 Отримання з прикладної схеми ISO 19109 в UML**

1) Мета тесту: якщо прикладна схема стандарту ISO 19109 в UML відповідає вимогам, викладеним в А.1.2, переконатися, що GML прикладна схема була правильно отримана з UML прикладної схеми.

2) Метод тесту: співставити описи обох прикладних схем та перевірити, чи відповідає конвертація з UML схеми до XML схеми правилам конвертації в Д.2.4.

3) Посилання: ISO 19136:2007, Д.2.4.

4) тип тесту: тест на здатність.

### **A.1.3 Випадки тестів прикладних схем стандарту ISO 19109 в UML, конвертованих з GML прикладних схем**

#### **A.1.3.1 Дійсна GML прикладна схема**

1) Мета тесту: якщо GML прикладна схема конвертована в прикладну схему стандарту ISO 19109 в UML, то переконатися, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити GML прикладну схему на те, що вона була правильно побудована згідно з Е.2.1.

3) Посилання: ISO 19136:2007, Е.2.1.

4) Тип тесту: тест на спроможність.

#### **A.1.3.2 Конвертація в прикладну схему стандарту ISO 19109 в UML**

1) Мета тесту : якщо GML прикладна схема відповідає вимогам, викладеним в А.1.3, переконатися, що прикладна схема стандарту 19109 в UML була правильно отримана з GML прикладної схеми.

2) Метод тесту: співставити описи обох прикладних схем та перевірити, чи відповідає конвертація з XML схеми в UML правилам конвертації в Е.2.3.

3) Посилання: ISO 19136:2007, Е.2.3.

4) Тип тесту: тест на спроможність.

#### **A.1.4 GML прикладні схеми, що визначають географічні об'єкти та колекції географічних об'єктів**

1) Мета тесту: якщо GML прикладна схема визначає географічні об'єкти, то переконатися в тому, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML схема побудована згідно з підрозділом 21.3.

3) Посилання: ISO 19136: 2007, підрозділ 21.3.

4) Тип тесту: тест на спроможність.

#### **A.1.5 GML прикладні схеми, що визначають просторові геометрії**

1) Мета тесту: якщо GML прикладна схема визначає просторові геометричні об'єкти, переконатися, що прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.4.

3) Посилання: ISO 19136:2007, підрозділ 21.4.

4) Test Type: тест на спроможність.

#### **A.1.6 GML прикладні схеми, що визначають просторові топології**

1) Мета тесту: якщо GML прикладна схема визначає об'єкти просторової топології, переконатися, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.5.

3) Посилання: ISO 19136:2007, підрозділ 21.5.

4) Тип тесту: тест на спроможність.

#### **A.1.7 GML прикладні схеми, що визначають час**

1) Мета тесту: якщо GML прикладна схема визначає часові об'єкти, переконатися, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.6.

3) Посилання: ISO 19136:2007, підрозділ 21. 6.

4) Тип тесту: тест на спроможність.

#### **A.1.8 GML прикладні схеми, що визначають референцні системи координат**

1) Мета тесту: якщо GML прикладна схема визначає об'єкти референціальних систем координат, то переконатися, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.7.

3) Посилання: ISO 19136:2007, підрозділ 21.7

4) Тип тесту: тест на спроможність.

**A.1.9 GML прикладні схеми, що визначають покриття**

1) Мета тесту: якщо GML прикладна схема визначає географічні об'єкти-покриття, переконатися, що GML прикладна схема була правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.8.

3) Посилання: ISO 19136:2007, підрозділ 21.8.

4) Тип тесту: тест на спроможність.

**A.1.10 GML прикладні схеми, що визначають спостереження**

1) Мета тесту: якщо GML прикладна схема визначає географічний об'єкт-спостереження, то переконатися, що GML прикладна схема була правильно побудована.

2) Метод тестування: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.9.

3) Посилання: ISO 19136:2007, підрозділ 21.9.

4) Тип тесту: тест на спроможність.

**A.1.11 GML прикладні схеми, що визначають словники та визначення**

1) Мета тесту: якщо GML прикладна схема визначає об'єкти-словники та об'єкти-визначення, то переконатися в тому, що GML прикладна схема правильно побудована.

2) Метод тесту: перевірити, чи була GML прикладна схема побудована згідно з підрозділом 21.10.

3) Посилання: ISO 19136:2007, підрозділ 21.10.

4) Тип тесту: тест на спроможність.

**A.1.12 GML прикладні схеми, що визначають значення**

1) Мета тесту: якщо GML прикладна схема визначає об'єкти значень, то переконатися, що GML прикладна схема була правильно побудована.

2) Метод тестування: перевірити, чи була прикладна схема GML побудована згідно з підрозділом 21.11.

3) Посилання: ISO 19136:2007, підрозділ 21.11.

4) Тип тесту: тест на спроможність.

**A.2 Абстрактний комплект тестів для GML профілів:****A.2.1 Дійсний GML профіль**

1) Мета тесту: перевірити, чи відповідає профіль правилам та інструкціям, викладеним у розділі 20 та підрозділі 21.12.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, розділ 20, підрозділ 21.12.

4) Тип тесту: тест на спроможність.

**A.2.2 Геометричні примітиви (просторові)****A.2.2.1 Типи даних для геометричних примітивів****A.2.2.1.1 Типи даних для нульвимірної геометрії**

1) Мета тесту: перевірити, чи включає GML профіль `gml:Point` та `gml:PointPropertyType`; якщо GML профіль включає також `gml:MultiPoint`, перевірити, що він включає `gml:MultiPointPropertyType`; перевірити, чи всі не скасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136, підрозділ 10.3, пункт 11.3.2.

4) Тип тесту: тест на спроможність.

#### **A.2.2.1.2 Типи даних для одновимірної геометрії**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.1 та включає `gml:Curve`, `gml:LineStringSegment`, `gml:LineString` та `gml:CurvePropertyType`; якщо GML профіль включає також `gml:MultiCurve`, перевірити, чи включає він `gml:MultiCurvePropertyType`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.1, пункти 10.4.1, 10.4.2, 10.4.7, 10.4.7.4-10.4.7.21, 11.3.3.

4) Тип тесту: тест на спроможність.

#### **A.2.2.1.3 Типи даних для двохвимірної геометрії**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.2 та включає `gml:Surface`, `gml:PolygonPatch`, `gml:Polygon`, `gml:SurfacePropertyType`, `gml:LinearRing` та `gml:Ring`; якщо GML профіль включає також `gml:MultiSurface`, перевірити, що він включає `gml:MultiSurfacePropertyType`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.2, пункти 10.5.1, 10.5.2, 10.5.4-10.5.9, 10.5.10, 11.3.4, підпункти 10.5.11.1, 10.5.12.4-10.5.11.6.

4) Тип тесту: тест на спроможність.

#### **A.2.2.1.4 Типи даних для трьохвимірної геометрії**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.3 та включає `gml:Solid`, `gml:SolidPropertyType` та `gml:Shell`; якщо GML профіль включає також `gml:MultiSolid`, перевірити, що він включає `gml:MultiSolidPropertyType`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.3, пункти 10.6.1, 10.6.2, 10.6.4-10.6.6, 11.3.5.

4) Тип тесту: тест на спроможність.

### **A.2.3 Геометричні комплекси (просторові)**

#### **A.2.3.1 Типи даних для геометричних комплексів**

##### **A.2.3.1.1 Типи даних для одновимірних геометричних комплексів**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.2 та включає `gml:CompositeCurve`, `gml:OrientableCurve`, `gml:GeometricComplex` та `gml:GeometricComplexPropertyType`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.2, підпункти 11.2.1.1, 11.2.1.1, 11.2.2.1, 11.2.2.2.

4) Тип тесту: тест на спроможність.

##### **A.2.3.1.2 Типи даних для двохвимірних геометричних комплексів**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.3 і A.2.3.1.1 та включає `gml:CompositeSurface` та `gml:OrientableSurface`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.3, A.2.3.1.1, пункт 10.5.11, підпункт 11.2.2.3.

4) Тип тесту: тест на спроможність.

#### **A.2.3.1.3 Типи даних для трьохвимірних геометричних комплексів**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.2.1.4 і A.2.3.1.2 та включає `gml:CompositeSolid`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.4, A.2.3.1.2, підпункт 11.2.2.4.

4) Тип тесту: тест на спроможність.

#### **A.2.4 Топологічні комплекси (просторові)**

##### **A.2.4.1 Типи даних для топологічних комплексів**

###### **A.2.4.1.1 Типи даних для одновимірних топологічних комплексів**

1) Мета тесту: перевірити, чи включає GML профіль `gml:TopoComplex`, `gml:TopoComplexMemberType`, `gml:Node`, `gml:directedNode`, `gml:Edge`, та `gml:directedEdge`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1, за виключенням `gml:pointProperty` у `gml:Node` та `gml:curveProperty` у `gml:Edge`.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділ 13.2 та 13.5, пункти 13.3.1, 13.3.2, 13.3.3.

4) Тип тесту: тест на спроможність.

###### **A.2.4.1.2 Типи даних для двохвимірних топологічних комплексів**

1) Мета тесту: перевірити, чи задовольняє GML-профіль вимоги A. 2.4.1.1 та включає `gml:Face` та `gml:directedFace`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1, за виключенням `gml:surfaceProperty` у `gml:Surface`.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A. 2.4.1.1, пункт 13.3.4.

4) Тип тесту: тест на спроможність.

###### **A.2.4.1.3 Типи даних для трьохвимірних топологічних комплексів**

1) Мета тесту: перевірити, чи задовольняє GML-профіль вимоги A. 2.4.1.2 та включає `gml:TopoSolid` та `gml:directedTopoSolid`; перевірити, чи всі нескасовані властивості елементів об'єктів є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1, за виключенням `gml:solidProperty` у `gml:TopoSolid`.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A. 2.4.1.2, пункт 13.3.5.

4) Тип тесту: тест на спроможність.

**A.2.5 Топологічні комплекси з геометричною реалізацією (просторові)****A.2.5.1 Типи даних для топологічних комплексів з геометричною реалізацією****A.2.5.1.1 Типи даних для одновимірних топологічних комплексів з геометричною реалізацією**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.2.1.1, A.2.2.1.2 та A.2.4.1.1 і включає властивості `gml:pointProperty` у `gml:Node` та `gml:curveProperty` у `gml:Edge` із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.2.1.1, A.2.2.1.2, A.2.4.1.1, пункти 13.3.2, 13.3.3.

4) Тип тесту: тест на спроможність.

**A.2.5.1.2 Типи даних для двохвимірних топологічних комплексів з геометричною реалізацією**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.2.1.3, A.2.4.1.2 та включає властивість `gml:surfaceProperty` у `gml:Face` із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A. 2.2.1.3, A.2.4.1.2, пункт 13.3.4.

4) Тип тесту: тест на спроможність.

**A.2.5.1.3 Типи даних для трьохвимірних топологічних комплексів з геометричною реалізацією**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.2.1.4, A.2.4.1.3 та включає властивість `gml:solidProperty` у `gml:ToroSolid` із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A. 2.2.1.4, A.2.4.1.3, пункт 13.3.5.

4) Тип тесту: тест на спроможність.

**A.2.6 Референцні системи координат**

1) Мета тесту: перевірити, чи містить GML профіль усі компоненти схеми, визначеної в розділі 12, які є обов'язковими або обов'язковими за певних умов згідно з ISO 19111.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 1913:2007, розділ 12, ISO 19111:2007, таблиці 1-41 та, у випадку проєційованих референцних систем координат – таблиці 42-56.

4) Тип тесту: тест на спроможність.

**A.2.7 Координатні операції**

1) Мета тесту: перевірити, чи містить GML профіль усі компоненти схеми, визначеної в підрозділі 12.6, які є обов'язковими або обов'язковими за певних умов згідно з ISO 19111.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділ 12.6, ISO 19111:2007, таблиці 42 – 56.

4) Тип тесту: тест на спроможність.

**A.2.8 Часова геометрія****A.2.8.1 Типи даних для нульвимірної геометрії**



1) Мета тесту: перевірити, чи включає GML профіль `gml:TimeInstant` та `gml:TimeInstantPropertyType`; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підпункти 14.2.1.1, 14.2.1.2, 14.2.1.4, 14.2.2.2, 14.2.2.3, 14.2.2.4, 14.2.2.7.

4) Тип тесту: тест на спроможність.

#### **A.2.8.2 Типи даних для одновимірної геометрії**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.8.1 та включає `gml:TimePeriod` та `gml:TimePeriodPropertyType`; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.8.1, підпункти 14.2.2.5, 14.2.2.6.

4) Тип тесту: тест на спроможність.

#### **A.2.9 Часова топологія**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.8.2 та включає `gml:TimeNode`, `gml:TimeNodePropertyType`, `gml:TimeEdge`, `gml:TimeEdgePropertyType`, `gml:TimeTopologyComplex` та `gml:TimeTopologyComplexType`; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, пункт 14.3.2.

4) Тип тесту: тест на спроможність.

#### **A.2.10 Часові референцні системи**

1) Мета тесту: перевірити, чи включає GML профіль `gml:TimeReferenceSystem` та хоча б одне з `gml:TimeCoordinateSystem`, `gml:TimeCalendar`, `gml:TimeClock`, `gml:TimeOrdinalReferenceSystem`; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділ 14.4.

4) Тип тесту: тест на спроможність.

#### **A.2.11 Динамічні географічні об'єкти**

1) Мета тесту: перевірити, чи задовольняє GML профіль всі вимоги A.2.8.2 та включає `gml:DynamicFeature`, `gml:AbstractTimeSlice`, `gml:history`. Перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (`maxOccurs`) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.8.2, підрозділ 14.5.

4) Тип тесту: тест на спроможність.

#### **A.2.12 Словники**

1) Мета тесту: перевірити, чи включає GML профіль gml:Dictionary; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, розділ 15.

4) Тип тесту: тест на спроможність.

#### **A.2.13 Словники одиниць**

1) Мета тесту: перевірити, чи включає GML профіль gml:UnitDictionary, gml:BaseUnit, gml:DerivedUnit, та gml:ConventionalUnit; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділ 16.2.

4) Тип тесту: тест на спроможність.

#### **A.2.14 Спостереження**

1) Мета тесту: перевірити, чи включає GML профіль gml:Observation та gml:DirectedObservation; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділи 17.1, 18.2.

4) Тип тесту: тест на спроможність.

#### **A.2.15 Покриття**

##### **A.2.15.1 Абстрактне покриття**

1) Мета тесту: перевірити, чи включає GML профіль gml:AbstractCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, підрозділ 16.4, пункти 19.3.1, 19.3.4-19.3.21.

4) Тип тесту: тест на спроможність.

##### **A.2.15.2 Дискретне точкове покриття**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:MultiPointCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункт 19.3.16.

4) Тип тесту: тест на спроможність.

##### **A.2.15.3 Дискретне лінійне покриття**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:MultiCurveCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункт 19.3.17.

4) Тип тесту: тест на спроможність.

#### **A.2.15.4 Дискретне покриття поверхонь**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:MultiSurfaceCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункт 19.3.18.

4) Тип тесту: тест на спроможність.

#### **A.2.15.5 Дискретне покриття тіл**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:MultiSurfaceCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункт 19.3.19.

4) Тип тесту: тест на спроможність.

#### **A.2.15.6 Grid-покриття**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:GridCoverage та gml:RectifiedGridCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункти 19.3.20, 19.3.21.

4) Тип тесту: тест на спроможність.

#### **A.2.15.7 Безперервне покриття**

1) Мета тесту: перевірити, чи задовольняє GML профіль вимоги A.2.15.1 та включає gml:AbstractContinuousCoverage; перевірити, чи всі нескасовані властивості елементів об'єкту є частиною профілю із значенням максимальної множинності (maxOccurs) не менше ніж 1.

2) Метод тестування: перевірити профіль.

3) Посилання: ISO 19136:2007, A.2.15.1, пункт 19.3.3.

4) Тип тесту: тест на спроможність.

### **A.3 Абстрактний комплект тестів для GML документів**

#### **A.3.1 Існування посилання на застосовну GML прикладну схему**

1) Мета тесту: перевірити існування посилання на GML прикладну схему, яку можна застосувати до GML документу.

2) Метод тестування: перевірити, чи є посилання на файл XML схеми, яким подано GML прикладну схему, в атрибуті xsi:schemaLocation кореневого елемента GML документу.

3) Посилання: ISO 19136:2007, підрозділ 21.1.

4) Тип тесту: базовий тест.

#### **A.3.2 Існування GML прикладної схеми, на яку спрямоване посилання**

1) Мета тесту: перевірити існування GML прикладної схеми, яку можна застосувати до GML документу.

2) Метод тестування: перевірити, чи є доступним файл XML схеми, яким подано GML прикладну схему, на яку посилається GML документ; перевірити також, чи всі документи, що мають бути доступними прямо чи непрямо через файл, на який спрямоване посилання, дійсно доступні.

3) Посилання: ISO 19136:2007, підрозділ 21.1.

4) Тип тесту: базовий тест.

### **A.3.3 Відповідність GML прикладної схеми, на яку спрямоване посилання**

1) Мета тесту: перевірити, чи відповідає цьому міжнародному стандарту GML прикладна схема, на яку посилається GML документ.

2) Метод тестування: перевірити, чи пройшла прикладна схема всі застосовні тести, визначені в A.1.

3) Посилання: ISO 19136:2007, A.1.

4) Тип тесту: тест на спроможність.

### **A.3.4 Дійсний XML**

1) Мета тесту: перевірити дійсність GML документу згідно до компонентів XML схеми GML прикладної схеми, яка відповідає вимогам цього міжнародного стандарту.

2) Метод тестування: перевірити GML документ на відповідність GML прикладної схеми, на яку спрямоване посилання; при цьому можна використовувати доречний програмний інструмент перевірки або виконувати її вручну з перевіркою всіх доречних визначень із специфікації XML схеми.

3) Посилання: ISO 19136:2007, 21.1.

4) Тип тесту: тест на спроможність.

### **A.3.5 Відповідність GML документу**

1) Мета тесту: перевірити, чи узгоджений GML документ з усіма іншими обмеженнями, визначеними у цьому міжнародному стандарті.

2) Метод тестування: перевірити, чи дотримані вимоги з додатків A.3.1-A.3.3, а GML документ задовольняє вимоги A.3.4, та чи узгоджений він з усіма іншими обмеженнями, визначеними у цьому міжнародному стандарті.

3) Посилання: ISO 19136:2007, розділи 7-21, особливо підпункти 7.2.3.4, 10.1.3.2, 10.1.3.3, 10.1.4.2, 16.4.11.

4) Тип тесту: тест на спроможність.

## **ДОДАТОК Б (нормативний)**

### **АБСТРАКТНИЙ КОМПЛЕКТ ТЕСТІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **Б.1 Випадки тестів на дотримання обов'язкових вимог відповідності**

##### **Б.1.1 GML профіль**

1) Мета тесту: перевірити, чи був GML профіль задокументований, тобто чи повністю він підтримується шляхом реалізації програмного забезпечення.

2) Метод тестування: перевірити документацію з реалізації програмного забезпечення для ідентифікації профілю; перевірити профіль на відповідність вимогам абстрактного комплексу тестів у A.1.12; перевірити, чи повністю програмне

забезпечення підтримує профіль та семантику, асоційовану з усіма компонентами схеми в профілі.

3) Посилання: ISO 19136:2007, A.1.12.

4) Тип тесту: тест на спроможність.

#### **Б.1.2 Підтримка локальних простих Xlink посилань**

1) Мета тесту: якщо реалізація програмного забезпечення здатна обробляти елементи GML об'єктів у форматі XML, перевірити, чи підтримує програмне забезпечення посилання на інші об'єкти у тому ж самому GML документі.

2) Метод тестування: перевірити, чи може реалізація програмного забезпечення обробляти екземпляри властивості, які використовують атрибут `xlink:href` із контентом `shorthand` указування на ресурс у тому ж самому XML документі.

3) Посилання: ISO 19136:2007 підрозділ 8.1.

4) Тип тесту: тест на спроможність.

#### **Б.1.3 Референцні системи координат, які використовують у географічних об'єктах (реалізація в програмному забезпеченні)**

1) Мета тесту: якщо реалізація програмного забезпечення здатна обробляти елементи GML об'єктів у форматі XML і якщо GML профіль реалізації включає географічні об'єкти, перевірити, чи дотриманий механізм призначення референцної системи координат за умовчанням для всіх геометричних об'єктів у географічному об'єкті.

2) Метод тестування: перевірити реалізацію програмного забезпечення, чи використаний атрибут `srsName`, що належить до `gml:Envelope`, тобто чи використане значення властивості `gml:boundedBy` об'єкту як референцна система координат за умовчанням для всіх геометричних об'єктів, закодованих шляхом включення в елементі географічного об'єкту.

3) Посилання: ISO 19136:2007, пункт 9.10.

4) Тип тесту: тест на спроможність.

### **Б.2 Випадки тестів дотримання необов'язкових вимог відповідності для реалізації програмного забезпечення із здатністю обробляти GML об'єкти у XML форматі**

#### **Б.2.1 Підтримка віддалених простих Xlink**

1) Мета тесту: перевірити, чи підтримує програмне забезпечення посилання на інші об'єкти у тому ж самому GML документі або поза ним.

2) Метод тестування: перевірити, чи задовольняє програмне забезпечення вимоги А.1.1.1 та може обробляти екземпляри властивості, які використовують атрибут `xlink:href` із контентом указування `Xpointer` на ресурс поза тим же самим XML документом.

3) Посилання: ISO 19136:2007, підрозділ 8.1.

4) Тип тесту: тест на спроможність.

#### **Б.2.2 Підтримка розширених Xlink посилань**

1) Мета тесту: перевірити, чи підтримує програмне забезпечення розширені Xlink посилання.

2) Метод тестування: перевірити, чи може програмне забезпечення обробляти атрибути розширених Xlink посилань.

3) Посилання: ISO 19136:2007, підрозділ 8.1.

4) Тип тесту: тест на спроможність.

### **Б.2.3 Підтримка для властивостей, які можна обнулити**

1) Мета тесту: перевірити, чи підтримує програмне забезпечення властивості, які можна обнулити.

2) Метод тестування: перевірити, чи може програмне забезпечення обробляти GML прикладні схеми з деклараціями елементу властивості з атрибутами  `xsi:nil`  та обробляти екземпляри з атрибутами  `xsi:nil`  та  `gml:nilReason`  у цих елементах.

3) Посилання: ISO 19136:2007, підпункти 8.2.3.1 та 8.2.3.2.

4) Тип тесту: тест на спроможність.

### **Б.2.4 Підтримка одиниць вимірювання**

1) Мета тесту: перевірити, чи може програмне забезпечення здійснювати перетворення між двома одиницями одного виду.

2) Метод тестування: перевірити, чи може програмне забезпечення обробляти значення атрибуту  `uom` , що належить до  `gml:MeasureType` , як це визначено в підпункті 7.3.3.7 та перетворювати міри до інших одиниць того ж виду з використанням словника одиниць, як визначено в підрозділі 16.2.

3) Посилання: ISO 19136:2007, підпункт 8.2.3.6, підрозділ 16.2.

4) Тип тесту: тест на спроможність.

### **Б.2.5 Підтримка семантики власності властивостей**

1) Мета тесту: перевірити, чи підтримує програмне забезпечення атрибут  `"owns"` .

2) Метод тестування: якщо програмне забезпечення може видаляти об'єкти з GML документу, перевірити, чи видаляє воно всі об'єкти, якими володіє інший об'єкт, як указано в атрибуті  `"owns"` , якщо цей об'єкт видалити.

3) Посилання: ISO 19136:2007, підпункт 7.2.3.5.

4) Тип тесту: тест на спроможність.

### **Б.2.6 Властивості метаданих**

1) Мета тесту: перевірити, чи всі властивості, де значенням є метадані про об'єкт, ідентифіковані як властивості метаданих.

2) Метод тестування: перевірити, чи всі елементи властивості, модель контенту яких отримана з  `gml:AbstractMetadataPropertyType` , ідентифіковані як властивості метаданих.

3) Посилання: ISO 19136:2007, підпункт 7.2.6.

4) Тип тесту: тест на спроможність.

### **Б.2.7 Підтримка GML профілів в перевірці екземпляру**

1) Мета тесту: перевірити, чи може програмне забезпечення використовувати GML профілі для перевірки екземпляру.

2) Тест методу: перевірити, чи використовує програмне забезпечення GML профілі для перевірки екземпляру, якщо на профіль спрямоване посилання з прикладної схеми з використанням анотації  `gml:gmlProfileSchema` .

3) Посилання: ISO 19136:2007, підрозділ 20.5

4) Тип тесту: тест на спроможність.

## **Б.3 Випадки тестів для написання GML**

### **Б.3.1 Можливість серіалізації**

1) Мета тесту: перевірити існування операції серіалізації у реалізації.

2) Метод тестування: перевірити програмного забезпечення та його документацію, чи забезпечують вони операцію серіалізації, що записує дійсні екземпляри GML об'єктів у форматі XML.

3) Посилання: ISO 19136:2007, підрозділ А.3.

4) Тип тесту: базовий тест.

### **Б.3.2 Дійсність серіалізації**

1) Мета тесту: перевірити, чи відповідає результат операції серіалізації цьому міжнародному стандарту.

2) Метод тестування: написати типовий GML документ із використанням операції серіалізації та перевірити, чи дійсні GML об'єкти у форматі XML.

3) Посилання: ISO 19136:2007, А.3.

4) Тип тесту: тест на спроможність.

### **Б.4 Випадок тесту читання GML**

1) Мета тесту: якщо програмне забезпечення може створювати об'єкти реалізації з GML елементів об'єкту в XML та серіалізувати ці об'єкти реалізації назад у GML об'єкти в форматі XML, перевірити, що воно робить це правильно.

2) Метод тестування: перевірити, чи дають послідовні дії створення об'єкту та серіалізації (див. Б.1.2) результат без втрати інформації.

3) Посилання: ISO 19136:2007, А.3.

4) Тип тесту: тест на спроможність.

### **Б.5 Випадки тестів для написання GML прикладних схем**

#### **Б.5.1 Здатність на серіалізацію**

1) Мета тесту: перевірити існування операції серіалізації у програмному забезпеченні.

2) Метод тестування: перевірити програмне забезпечення та його документацію, чи забезпечує воно операцію серіалізації, яка створює дійсні екземпляри GML прикладних схем у форматі XML схеми.

3) Посилання: ISO 19136:2007, А.1.

4) Тип тесту: базовий тест.

#### **Б.5.2 Дійсність серіалізації**

1) Мета тесту: перевірити, що результат операції серіалізації відповідає цьому міжнародному стандарту.

2) Метод тестування: створити типові GML прикладні схеми з використанням операції серіалізації та перевірити, чи відповідають GML прикладні схеми GML профілю реалізації програмного забезпечення цьому міжнародному стандарту.

3) Посилання: ISO 19136:2007, А.1, Б.1.1.

4) Тип тесту: тест на спроможність.

### **Б.6 Випадки тестів для читання GML прикладних схем**

1) Мета тесту: якщо реалізація може створювати об'єкти реалізації з GML прикладної схеми у XML та серіалізувати ці об'єкти реалізації назад у GML прикладні схеми у форматі XML, перевірити, що воно робить це правильно;

2) Метод тестування: перевірити чи дають послідовні дії створення об'єкту та серіалізація (див. Б.1.2) результат без втрати інформації;

3) Посилання: ISO 19136:2007, А.1, Б.5;

4) Тип тесту: тест на здатність.

ДОДАТОК В  
(довідковий)  
**GML СХЕМА**

Документи XML схеми з GML схемою можна знайти он-лайн за наступним URL на веб-сторінці ISO:

[http://www.iso.org/ittf/ISO\\_19136\\_Schemas](http://www.iso.org/ittf/ISO_19136_Schemas)

Документи схеми структуровані по модулях, як показано в додатку Л, тобто по таких документах схеми:

- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/basicTypes.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/basicTypes.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/coordinateOperations.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/coordinateOperations.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/coordinateReferenceSystems.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/coordinateReferenceSystems.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/coordinateSystems.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/coordinateSystems.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/coverage.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/coverage.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/datums.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/datums.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/dictionary.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/dictionary.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/direction.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/direction.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/dynamicFeature.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/dynamicFeature.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/feature.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/feature.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/geometryAggregates.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/geometryAggregates.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/geometryBasic0d1d.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/geometryBasic0d1d.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/geometryBasic2d.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/geometryBasic2d.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/geometryComplexes.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/geometryComplexes.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/geometryPrimitives.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/geometryPrimitives.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/gml.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/gml.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/gmlBase.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/gmlBase.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/grids.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/grids.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/measures.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/measures.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/observation.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/observation.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/referenceSystems.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/referenceSystems.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/temporal.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/temporal.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/temporalReferenceSystems.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/temporalReferenceSystems.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/temporalTopology.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/temporalTopology.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/topology.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/topology.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/units.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/units.xsd)
- [http://www.iso.org/ittf/ISO\\_19136\\_Schemas/valueObjects.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/valueObjects.xsd)

Додатковий документ [www.iso.org/ittf/ISO\\_19136\\_Schemas/defaultStyle.xsd](http://www.iso.org/ittf/ISO_19136_Schemas/defaultStyle.xsd) містить компоненти довідкової схеми.

Документ XML схеми посилань Xlink розташований на <http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd>.

Альтернативне сховище схем – веб-сторінка схем Відкритого геопросторового консорціуму:

<http://schemas.opengis.net/gml/3.2.1/>



ДОДАТОК Г  
(нормативний)  
**РЕАЛІЗОВАНИЙ ПРОФІЛЬ МІЖНАРОДНИХ СТАНДАРТІВ КОМПЛЕКСУ  
ISO 19100 ТА РОЗШИРЕНЬ**

### Г.1 Загальні зауваження

Загальне відношення між комплексом міжнародних стандартів ISO 19100 та GML розглянуто в розділі 6. У цьому додатку докладно описано профіль концептуальної моделі, визначеної в комплексі міжнародних стандартів ISO 19100 та реалізованої з допомогою GML (див. Г.2), а також розширення цього профілю (див. Г.3).

В цьому міжнародному стандарті “профіль” означає чистий піднабір.

### Г.2 Профіль комплексу ISO 19100, використаний у GML

#### Г.2.1 Огляд

У наступних підрозділах описано профіль комплексу міжнародних стандартів ISO 19100, який використовують у GML. В описі діаграм класів профілю розглянуто відношення та перехід у GML схему.

**Примітка 1.** В цілому правила кодування, розглянуті в додатку Д, були використані також у кодуванні GML схеми. Однак, оскільки GML схема була головне *handcrafted*, в ній задіяно спеціальні можливості середовища реалізації, тобто XML та XML схему. Прикладами є низка наперед визначених базових типів (прості чи комплексні типи з простим контентом) або використання глобальних елементів також для властивостей з тим, щоб зробити їх такими, якими можна замінити (наприклад, для визначення альтернативних імен для скасованих імен властивостей). Ці випадки задокументовано в наступних підрозділах або є безпосередніми.

Тільки елементи з міжнародних стандартів, розглянуті далі, є частиною цього профілю. Елементи з інших міжнародних стандартів не є частиною профілю.

В силу природи GML жодна операція будь-якого класу не є частиною профілю.

Через природу GML жодна операція будь-якого класу не входить до цього профілю.

Крім цього, класи інтерфейсу (стереотип <<Interface>>) без структури даних та відношень “Realization” до класів без структури даних були видалені.

Також, проходимість асоціацій була обмежена до напрямків, у яких GML подає явні властивості об’єктів (більшість асоціацій у GML схемі проходимі тільки в єдиному напрямку).

**Примітка 2.** Жодні скасовані типи, елементи та атрибути GML не розглянуті в цьому додатку.

**Примітка 3.** Загальні правила для переходу з UML до XML схеми для прикладних GML схем визначені в додатку Д.

**Примітка 4.** В цьому додатку простір імен "xsd:" використаний для посилань на простір імен XML схеми, яким є "<http://www.w3.org/2001/XMLSchema>". Простір імен “gml:” посилається на простір імен GML, яким є "<http://www.opengis.net/gml/3.2>".

У таблиці Г.1 подано перехід між пакетами високого рівня комплексу міжнародних стандартів ISO 19100 та підрозділами цього міжнародного стандарту, що визначають компоненти GML схеми, які реалізують типи з цих пакетів:

**Таблиця Г. 1** – Огляд впроваджених пакетів комплексу міжнародних стандартів ISO 19100

UML пакет	Префікс UML класу	Підрозділ ISO 19136	Підрозділ додатку Г
ISO/TS 19103:Базові типи: Одиниці виміру	-	8.2.3.6, 16.2	Г.2.2
ISO 19107:Геометрія: Геометричний корінь	GM	10.1.3	Г.2.3.2
ISO 19107:Геометрія: Геометричний примітив	GM	10.2, 10.1.3, 10.4, 10.5, 10.6	Г.2.3.3
ISO 19107:Геометрія: Геометричний комплекс	GM	11.2	Г.2.3.6
ISO 19107:Геометрія: Геометричні агрегати	GM	11.3	Г.2.3.5
ISO 19107:Геометрія: Координатна геометрія	GM	10.1.4	Г.2.3.4
ISO 19107:Топологія: Корінь топології	TP	13.2	Г.2.4.2
ISO 19107:Топологія: Примітив топології	TP	13.3	Г.2.4.3
ISO 19107:Топологія: Комплекс топології	TP	13.5	Г.2.4.4
ISO 19108:Часові об'єкти	TM	14.2, 14.3	Г.2.5.2 – Г.2.5.6
ISO 19108:Часові референсні системи	TM	14.4	Г.2.5.7
ISO 19111:SC_CoordinateReferenceSystem	SC	12.2, 12.3	Г.2.7.3
ISO 19111:SC_CoordinateSystem	CS	12.4	Г.2.7.4
ISO 19111:SC_Datum	CD	12.5	Г.2.7.5
ISO 19111:SC_CoordinateOperation	CC	12.5	Г.2.7.6
ISO 19123	CV	19	Г.2.11

У таблиці Г. 2 поданий перехід між концептуальними UML класами, реалізованими цим міжнародним стандартом, та асоційованими GML елементом об'єкту, типом XML схеми та типом GML властивості.

Таблиця складається з чотирьох стовпців. Для надання повного переходу від UML типу (перший стовпець) до XML схеми, як використано в GML, потрібні три різних переходи, показані в трьох інших стовпцях. Це є результатом відмінностей у переході загальної моделі географічного об'єкту в UML та XML схеми, головне тому, що XML схема відокремлює XML елементи та їх модель контенту.

Цю таблицю треба читати таким чином:

– У першому стовпці ("UML клас") перелічено класи стандартів комплексу ISO 19100, що реалізовані в GML схемі.

Іноді цей стовпець порожній ("—"), що означає, що GML елементи та типи з інших стовпців реалізують концепцію, яка не визначена в комплексі стандартів ISO 19100, але це реалізовано в Г.3.

– У другому стовпці ("GML елемент об'єкту") визначено GML елемент об'єкту, який реалізує тип.

Цю інформацію зокрема використовують у двох ситуаціях у правилах кодування в додатку Д:

- коли жодний наперед визначений тип властивості для елемента об'єкту не є частиною GML схеми (див. четвертий стовпець), і тип властивості треба створювати в прикладній схемі;
- коли підтип типу визначено в прикладній схемі, у випадку чого елемент об'єкту, яким подано підтип, треба визначати як частину групи заміни цього елемента об'єкту типу.

Там, де не існує жодного відповідного елемента об'єкту в GML схемі, це треба зазначити порожньою коміркою ("—").

- У третьому стовпці ("GML тип") визначено тип XML схеми, що визначає модель контенту GML елемента об'єкту у другому стовпці.

Цей тип XML схеми зокрема використовують у правилах кодування в додатку Д, коли підтип типу в першому стовпці змодельовано в прикладній схемі.

Коли жодного відповідного типу XML схеми не існує в GML схемі, це зазначено порожньою коміркою ("—").

- У четвертому стовпці ("GML тип властивості") визначено тип, який використовують як XML тип схеми, якщо цей тип використовують як значення властивості в прикладній схемі. У цьому випадку цей стовпець надає значення типу XML схеми, що є реалізацією цього типу в GML схемі.

*Приклад 1.* Якщо тип географічного об'єкту має властивість із типом "GM\_Point", то в поданні в XML схемі відповідний GML елемент властивості має gml:PointPropertyType в якості свого типу.

У випадку класу із стереотипом <<DataType>> не надають жодного подання властивості в XML схемі, оскільки типи даних, визначені в GML схемі, звичайно не призначені для використання як значення властивостей географічного об'єкту.

Іноді GML схема не містить наперед визначеного типу властивості для цього типу, і, якщо потрібно для прикладної схеми, тип властивості треба будувати згідно з правилами для GML типів властивості (див. 7.2.3), де GML елемент об'єкту наданий у другому стовпці того ж рядка.

Якщо значення анотоване з "(group)", то властивість реалізують шляхом посилання на глобальну групу, вказану в комірці замість локального елемента властивості.

Таблиця Г.2 – Реалізація типів комплексу міжнародних стандартів ISO 19100

UML клас	GML елемент об'єкту	GML тип	GML тип властивості
GM_Object	gml:AbstractGeometry	gml:AbstractGeometryType	gml:GeometryPropertyType
GM_Primitive	gml:AbstractGeometricPrimitive	gml:AbstractGeometricPrimitiveType	gml:GeometricPrimitiveProper
DirectPosition	—	—	gml:DirectPositionType
GM_Position	—	—	gml:geometricPositionGroup
GM_PointArray	—	—	gml:geometricPositionListGro
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	<i>anonymous property type</i> <sup>1</sup>
GM_TriangulatedSurfac	gml:TriangulatedSurface	gml:TriangulatedSurfaceType	<i>anonymous property type</i>
GM_Tin	gml:Tin	gml:TinType	<i>anonymous property type</i>
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	<i>anonymous property type</i>
GM_CompositeSurface	gml:CompositeSurface	gml:CompositeSurfaceType	<i>anonymous property type</i>
GM_CompositeSolid	gml:CompositeSolid	gml:CompositeSolidType	<i>anonymous property type</i>
GM_Complex	gml:GeometricComplex	gml:GeometricComplexType	gml:GeometricComplexPrope
GM_Aggregate	gml:MultiGeometry	gml:MultiGeometryType	gml:MultiGeometryPropertyT
GM_MultiPoint	gml:MultiPoint	gml:MultiPointType	gml:MultiPointPropertyType

UML клас	GML елемент об'єкту	GML тип	GML тип властивості
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	—
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	—

UML клас	GML елемент об'єкту	GML тип	GML тип властивості
GM_Triangle	gml:Triangle	gml:TriangleType	—
TP_Object	gml:AbstractTopology	gml:AbstractTopologyType	<i>anonymous property type</i>
TP_Node	gml:Node	gml:NodeType	gml:DirectedNodePropertyTy
TP_Edge	gml:Edge	gml:EdgeType	gml:DirectedEdgePropertyTy
TP_Face	gml:Face	gml:FaceType	gml:DirectedFacePropertyType
TP_Solid	gml:TopoSolid	gml:TopoSolidType	gml:DirectedTopoSolidProper
TP_DirectedNode	—	—	gml:DirectedNodePropertyTy
TP_DirectedEdge	—	—	gml:DirectedEdgePropertyTy
TP_DirectedFace	—	—	gml:DirectedFacePropertyType
TP_DirectedSolid	—	—	gml:DirectedTopoSolidProper
TP_Complex	gml:TopoComplex	gml:TopoComplexType	gml:TopoComplexPropertyTy
—	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	<i>anonymous property type</i>
TM_Complex	gml:AbstractTimeComplex	gml:AbstractTimeComplexType	<i>anonymous property type</i>
TM_GeometricPrimitive	gml:AbstractTimeGeometricPrimitive	gml:AbstractTimeGeometricPrimitiveType	gml:TimeGeometricPrimitiveP
TM_Instant	gml:TimeInstant	gml:TimeInstantType	gml:TimeInstantPropertyType
TM_Period	gml:TimePeriod	gml:TimePeriodType	gml:TimePeriodPropertyType
TM_TopologicalComple	gml:TimeTopologyComplex	gml:TimeTopologyComplexType	gml:TimeTopologyComplexP
TM_TopologicalPrimitiv	gml:AbstractTimeTopologyPrimitive	gml:AbstractTimeTopologyPrimitiveType	gml:TimeTopologyPrimitivePr
TM_Node	gml:TimeNode	gml:TimeNodeType	gml:TimeNodePropertyType
TM_Edge	gml:TimeEdge	gml:TimeEdgeType	gml:TimeEdgePropertyType
TM_PeriodDuration	—	—	gml:duration (property)
TM_IntervalLength	—	—	gml:timeInterval (group),
TM_Duration	—	—	gml:timeLength (group)
TM_Position	—	—	gml:TimePositionType

UML клас	GML елемент об'єкту	GML тип	GML тип властивості
TM_IndeterminateValue	—	—	@TimeIndeterminateValue
TM_Coordinate	—	—	xsd:decimal
TM_CalDate	—	—	gml:CalDate
TM_ClockTime	—	—	xsd:time
TM_DateAndTime	—	—	xsd:dateTime
TM_Calendar	gml:TimeCalendar	gml:TimeCalendarType	gml:TimeCalendarPropertyTy
TM_CalendarEra	gml:TimeCalendarEra	gml:TimeCalendarEraType	gml:TimeCalendarEraPropert
TM_Clock	gml:TimeClock	gml:TimeClockType	gml:TimeClockPropertyType
TM_CoordinateSystem	gml:TimeCoordinateSystem	gml:TimeCoordinateSystemType	<i>anonymous property type</i>
TM_OrdinalReferenceSy	gml:TimeOrdinalReferenceSystem	gml:TimeOrdinalReferenceSystemType	<i>anonymous property type</i>
TM_OrdinalEra	gml:TimeOrdinalEra	gml:TimeOrdinalEraType	gml:TimeOrdinalEraProperty
SC_CRS	gml:AbstractCRS	gml:AbstractCRSType	gml:CRSPropertyType
SI_LocationInstance	—	—	gml:LocationName
CV_Coverage	gml:AbstractCoverage	gml:AbstractCoverageType	<i>anonymous property type</i>
CV_ContinuousCoverag	gml:AbstractContinuousCoverage	gml:AbstractContinuousCoverageType	<i>anonymous property type</i>
CV_DiscreteCoverage	gml:AbstractDiscreteCoverage	gml:DiscreteCoverageType	<i>anonymous property type</i>
CV_DiscretePointCovera	gml:MultiPointCoverage	gml:MultiPointCoverageType	<i>anonymous property type</i>
CV_DiscreteCurveCover	gml:MultiCurveCoverage	gml:MultiCurveCoverageType	<i>anonymous property type</i>
CV_DiscreteSurfaceCov	gml:MultiSurfaceCoverage	gml:MultiSurfaceCoverageType	<i>anonymous property type</i>
CV_DiscreteSolidCovera	gml:MultiSolidCoverage	gml:MultiSolidCoverageType	<i>anonymous property type</i>
CV_DiscreteGridPointC	gml:GridCoverage	gml:GridCoverageType	<i>anonymous property type</i>
CharacterString	—	—	xsd:string
Boolean	—	—	xsd:boolean
Real, Number	—	—	xsd:double
Decimal	—	—	xsd:decimal
Date	—	—	xsd:date
Time	—	—	xsd:time
DateTime	—	—	xsd:dateTime

UML клас	GML елемент об'єкту	GML тип	GML тип властивості
Integer	—	—	xsd:integer, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:positiveInteger <sup>2</sup>
Vector	—	—	gml:VectorType
GenericName,	—	—	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
<p>1. Анонімний тип згідно до шаблону для GML типів властивості. Елемент об'єкту вбудований або такий, на який спрямоване посилання, є елементом у четвертому стовпці в тому ж самому рядку.</p> <p>2. Множинні значення у другому стовпці подано для підтримки зворотного перекладу, описаного в додатку E.</p>			



## Г.2.2 Мова концептуальної схеми ISO/TS 19103

У цьому підрозділі визначені базові типи, визначені в ISO/TS 19103, які є безпосередньо наявними в GML. Часто прості типи, визначені XML схемою, використовують напряду:

- “CharacterString”, реалізований з допомогою xsd:string; кодування характеру визначено в інструкції з обробки в XML документі (за умовчанням для XML документів приймають UTF-8);
- “Date”, реалізований з допомогою xsd:date;
- “DateTime”, реалізований з допомогою xsd:dateTime;
- “Time”, реалізований з допомогою xsd:time;
- “Real”, реалізований з допомогою xsd:double;
- “Decimal” взагалі реалізований з допомогою xsd:decimal; з практичних причин десяткові значення будуть часто подавати у схемах з допомогою xsd:double;
- загальний базовий тип “Number”, взагалі реалізований у GML схемі з допомогою xsd:double;
- “Integer”, реалізований з допомогою xsd:integer;
- “Boolean”, реалізований з допомогою xsd:Boolean;
- “Measure”, реалізований з допомогою gml:MeasureType; значення належить до типу xsd:double, uom-визначення реалізоване з допомогою URI, яке звичайно вирішує до елементу <gml:UnitDefinition> або добре відомого рядку одиниці вимірювання (див. пункт 8.2.3.6).

Одиницю вимірювання (“UnitOfMeasure”) реалізують із допомогою gml:UnitDefinitionType.

З допомогою GML такі реалізовано такі підтипи “Міри” (“Measure”), кожний з атрибутом uom, що вказує на відповідне визначення одиниці:

- “Length” → gml:LengthType;
- “Scale” → gml:ScaleType;
- “Area” → gml:AreaType;
- “Volume” → gml:VolumeType;
- “Speed” → gml:SpeedType;
- “Time” → gml:TimeType;
- “Angle” → gml:AngleType;
- “Vector” реалізовано з допомогою gml:VectorType;
- “GenericName“, “LocalName“ та “ScopedName“ реалізовано з допомогою gml:CodeType, де designator простору імен є URI;
- “Measure” реалізовано з допомогою gml:MeasureType, uom-атрибут є URI, що вказує на визначення одиниці;
- “UnitOfMeasure” реалізовано з допомогою gml:UnitDefinitionType.

**Примітка 1.** В ISO/TS 19103 вектор описано як “упорядкований набір чисел, які називають координатами, якими подано позицію в системі координат”. GML використовує вектор у цьому розумінні та надає спроможність явно вказувати систему координат, асоційовану з вектором.

ISO/TS 19103 визначає, що значення “NULL” еквівалентні. Цей міжнародний стандарт використовує більш явний підхід через надання механізму визначення причини для значення “nil”. Використання в аплікації цієї доданої інформації є необов’язковим.

### Г.2.3 Просторова схема (геометрія) стандарту ISO 19107

#### Г.2.3.1 Огляд

UML модель GML профілю, визначена в цьому додатку, описує концептуальну модель абстрактних типів, визначених в ISO 19107. Такі ж самі імена для класів та їх властивостей, як в ISO 19107, використовують для документування GML профілю для полегшення порівняння з цим стандартом.

**Примітка 1.** Див. подробиці в ISO 19107:2003, розділ 2.

Додаткові зміни, показані в таблиці Г.3, були застосовані до пакету геометрії ISO 19107.

**Таблиця Г.3.** Опис профілю стандарту ISO 19107 (геометрія)

Зміна	Пояснення
GM_Primitive: асоціацію “Interior to” видалено	Нині не підтримується в GML
GM_Polygon: атрибут “SpanningSurface” видалено	Нині не підтримується в GML
GM_Solid: операцію “boundary()” конвертовано в атрибут	Оскільки межа, що належить до GM_Solid, доступна тільки через операцію “boundary()”, додано атрибут такого ж імені. Значення атрибуту є результатом операції “boundary()”, як визначено в ISO 19107.
GM_Complex: асоціацію “Contains” видалено	Нині не підтримується в GML.
Похідні атрибути видалені в підтипах GM_MultiPrimitive	Ці атрибути можна отримати з цифрового подання об’єктів, тому зайву інформацію пропущено.
GM_CompositePoint видалено	GM_CompositePoint не додає жодної додаткової інформації. Цей тип додано в ISO 19107 тільки для повноти, але очікують, що його можна використовувати в документах екземплярів. Тому він пропущений у GML
GM_PolynomialSpline зроблено абстрактним	Нині не може мати екземпляри в GML, але підтип GM_CubicSpline може.
GM_LineSegment видалено	Не підтримується в GML, натомість треба використовувати GM_LineString з двома контрольними точками.
GM_CurveBoundary видалено	Використовують тільки в операціях.
GM_ComplexBoundary видалено	Використовують тільки в операціях

**Примітка 2.** OrientableCurve та GM\_OrientableSurface є “неабстрактними” (згідно до ISO 19107).

#### Г.2.3.2 Корінь геометрії

На діаграмах UML класів на рисунках Г.1 і Г.2 показано профіль пакету “корінь геометрії” (доречно порівняти з ISO 19107:2003, рисунки 5 і 6).

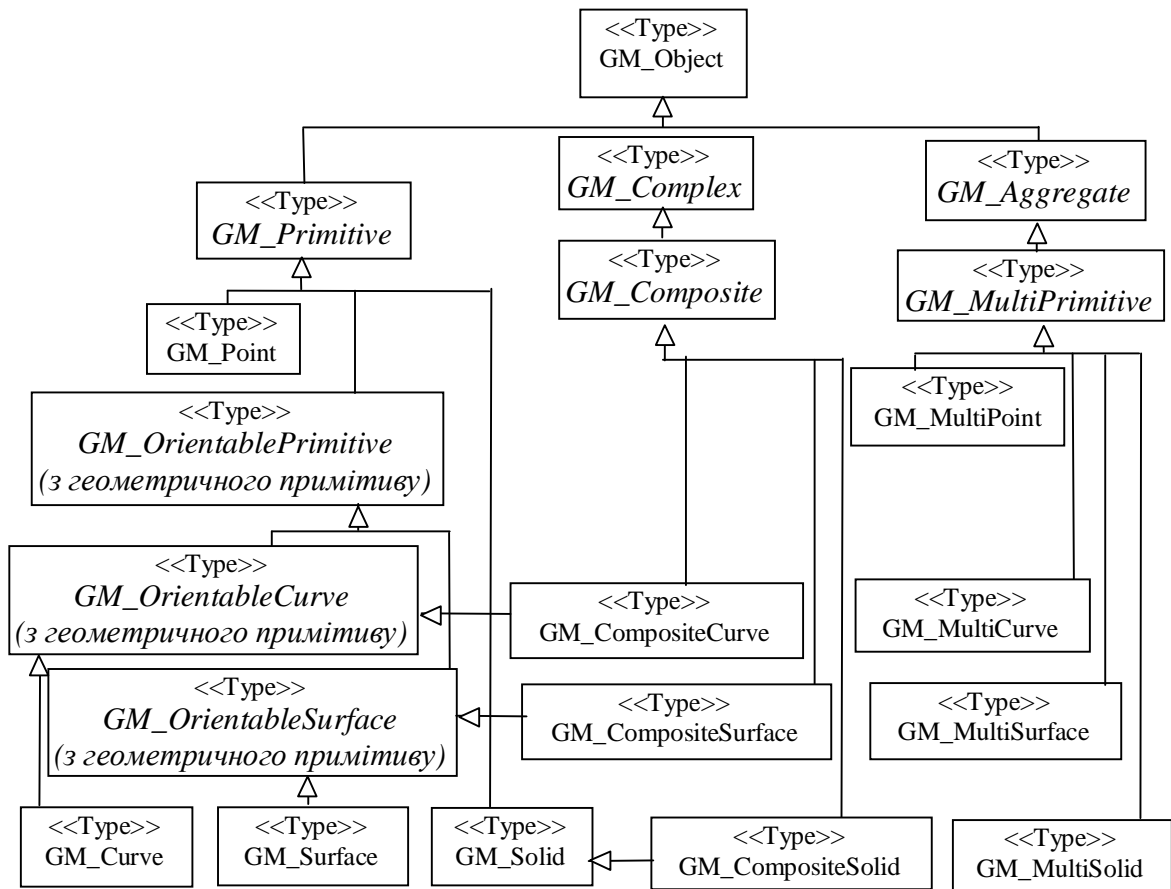


Рисунок Г.1 – Реалізовані підтипи геометричного об'єкту GM\_Object

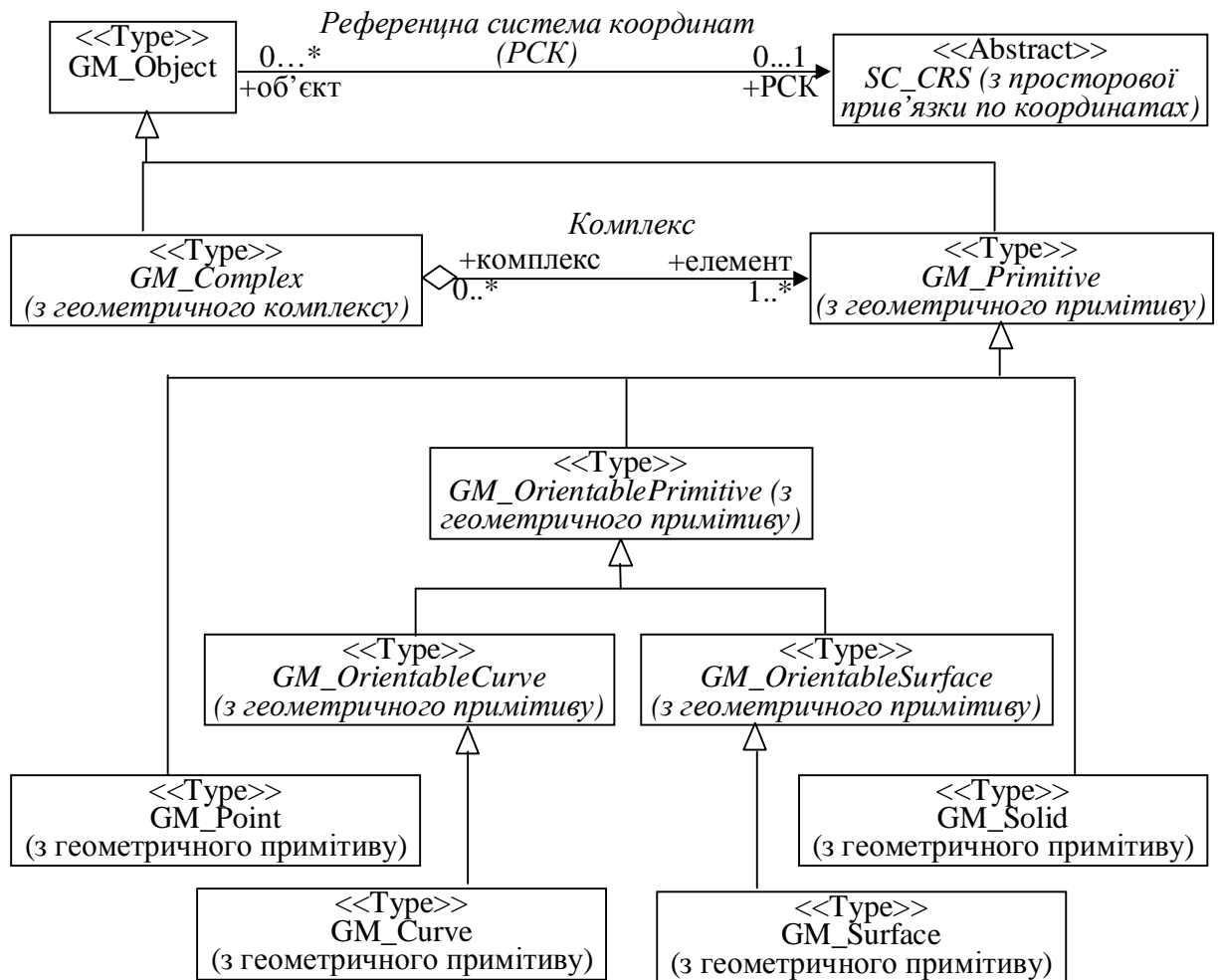


Рисунок Г.2 – Геометричні примітиви

Переклад різних класів у GML схему пояснено у наступних підрозділах, де показано деталі ієрархії класів.

“GM\_Object” подано елементом об’єкту “AbstractGeometry”, роль “PCK” (“CRS”) подана властивістю “srsName”.

Елемент “AbstractGeometry” може мати додаткові властивості: необов’язковий елемент “description”, нуль або більше елементів “name” та обов’язковий атрибут “gml:id”. Останній зокрема корисний у підтримці повторного використання елементів геометрії “через посилання”, з дотриманням компактного синтаксису зовнішніх покажчиків XPointer.

### Г.2.3.3 Примітив геометрії

На діаграмах UML класів на рисунках Г.3-Г.9 показано профіль пакету кореня геометрії “Geometry root” (доречно порівняти ISO 19107:2003, рисунки з 7 по 13).

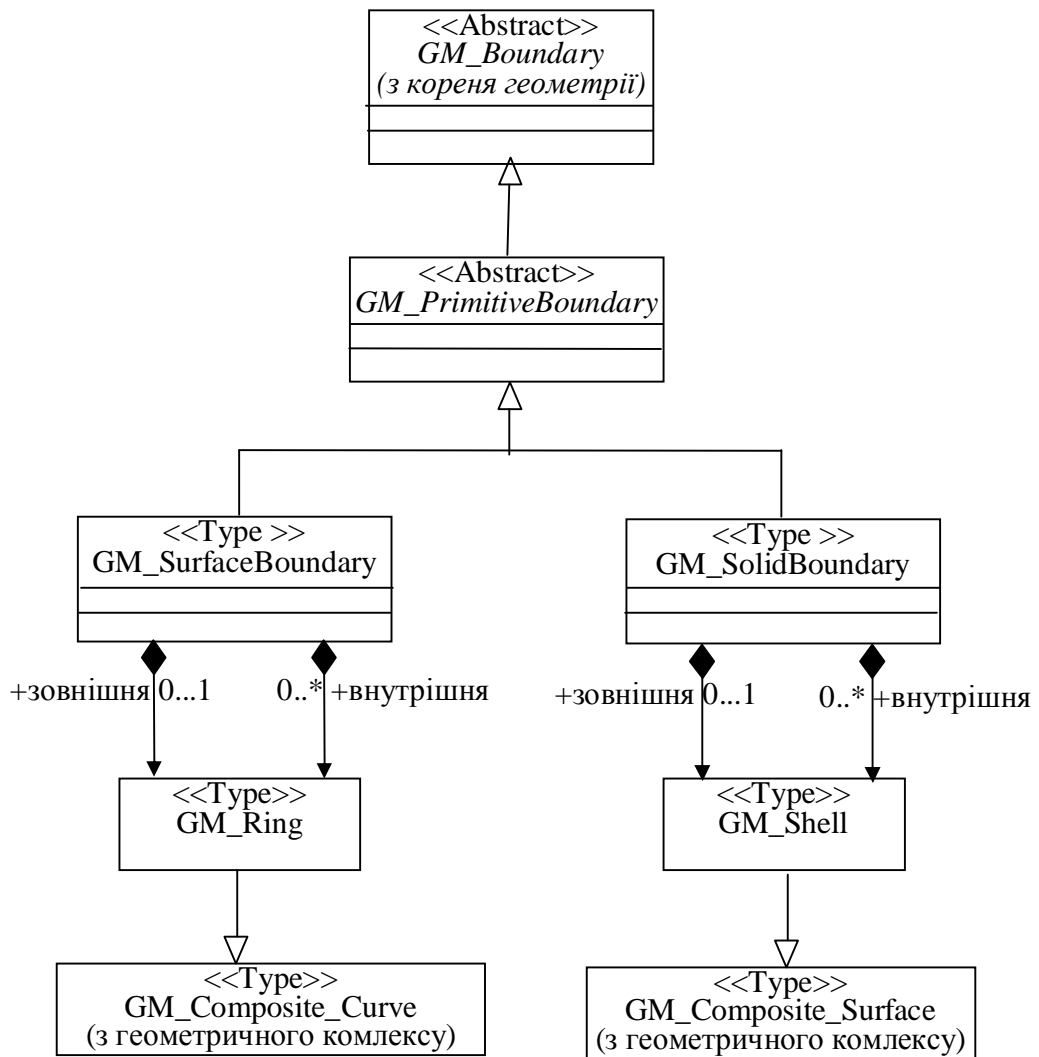


Рисунок Г.3 – Межі геометричних примітивів

Класи меж із стандарту ISO 19107 не подані явним чином у GML. У ISO 19107 типи меж звичайно повертають значення операції “boundary()”. Оскільки межі всіх ділянок поверхні або тіл треба явним чином подавати в GML як властивості, то властивості “exterior” та “interior” визначені у GML безпосередньо як властивості поверхні (ділянки) чи тіла.

“GM\_Ring” подають елементом об’єкту “Ring”. Тоді як “Ring” не може замінити “CompositeCurve” у GML, він структурно ідентичний композитній кривій.

“GM\_Shell” подають елементом об’єкту “Shell”. Оскільки “Shell” не може замінити “CompositeSurface” у GML, він структурно ідентичний до композитної поверхні.

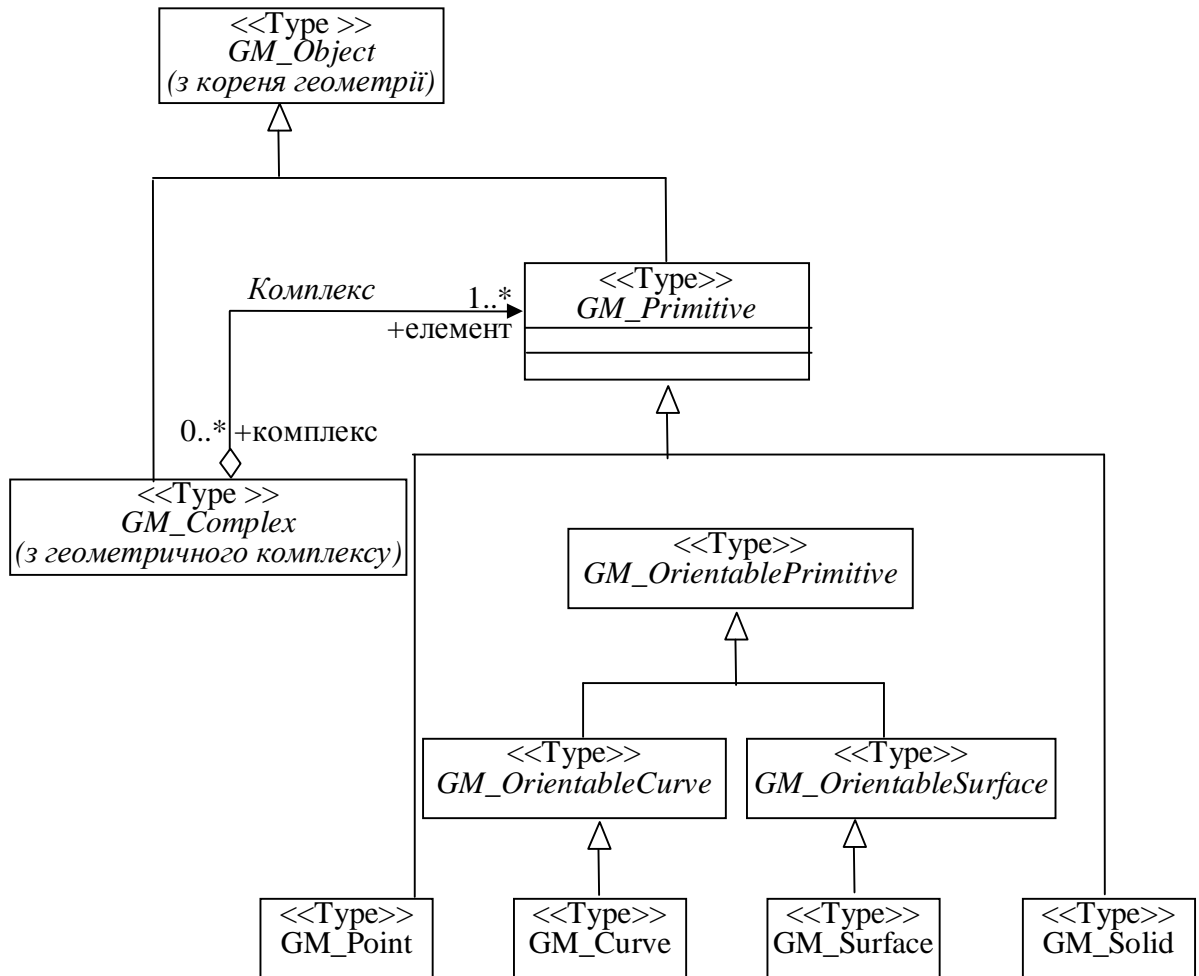


Рисунок Г.4 – Геометричні примітиви

“GM\_Primitive” поданий елементом об’єкту “AbstractGeometricPrimitive” (обидва як абстрактні). Роль “комплекс” не є проходимою в GML.

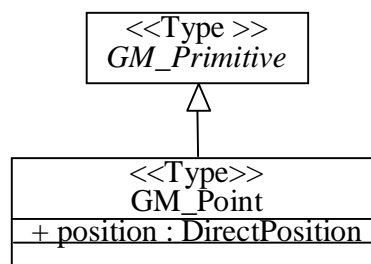


Рисунок Г.5 – Точка

“GM\_Point” подають елементом об’єкту “Point” у GML. Атрибут “position” подають властивістю “pos” (типом значення є “DirectPosition”).

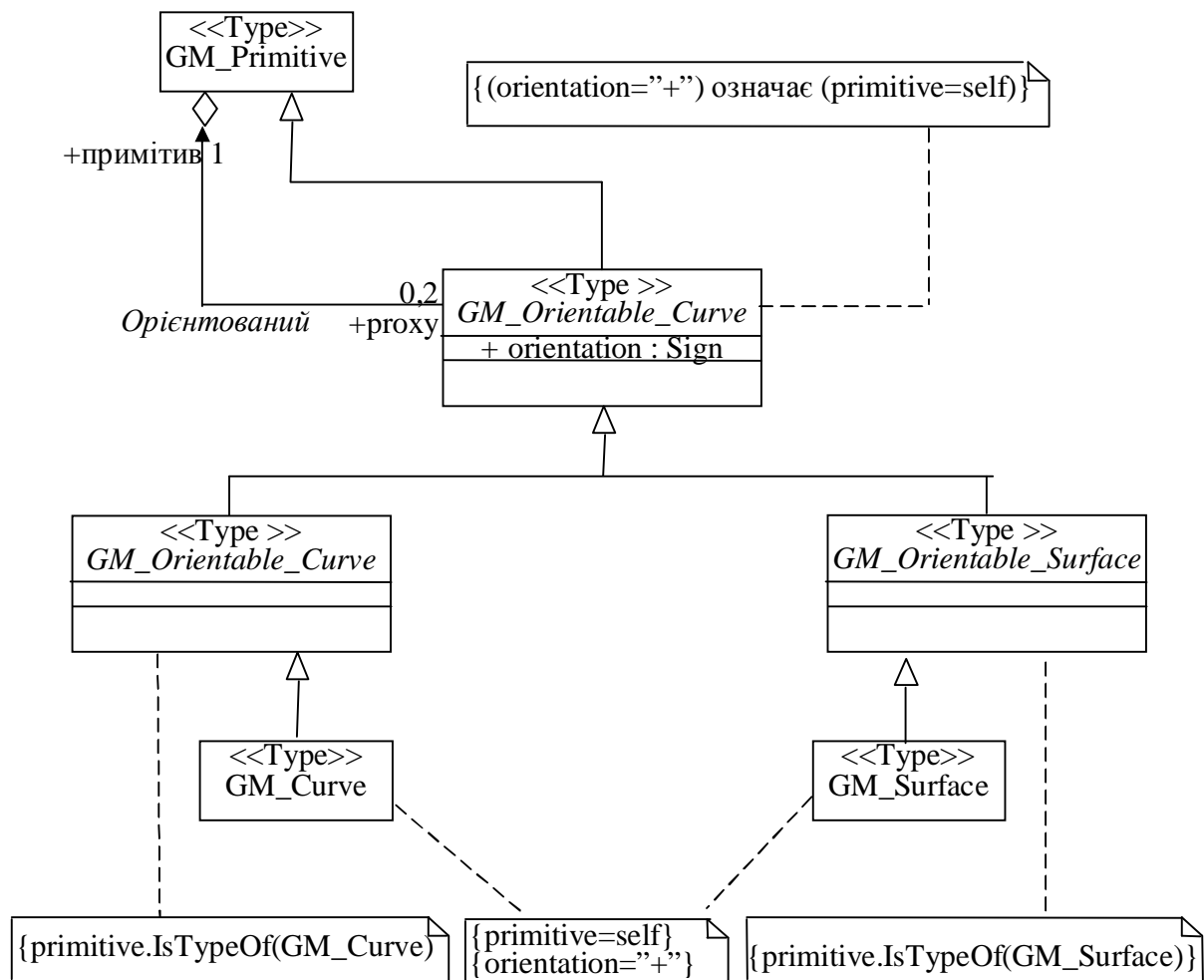


Рисунок Г.6 – Крива та поверхня

“GM\_Curve” подають елементом об’єкту “Curve” у GML. Орієнтація не є явною властивістю кривої “Curve” та неявно фіксована як “+”.

“GM\_OrientableCurve” подають елементом об’єкту “OrientableCurve” у GML. Роль “примітив” подають властивістю “baseCurve”.

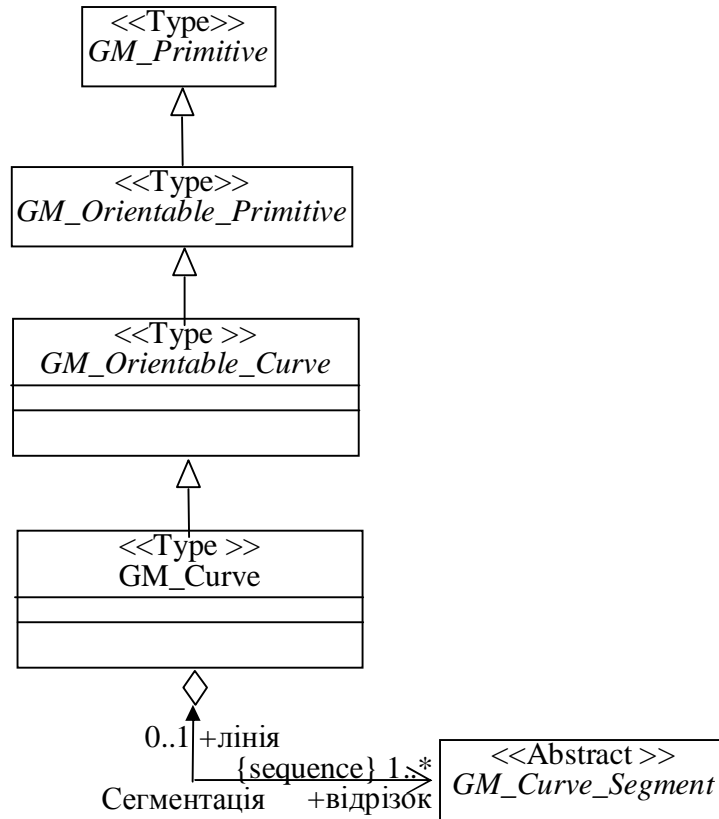
“GM\_Surface” подають елементом об’єкту “Surface” у GML. Орієнтація не є явною властивістю поверхні “Surface” та неявно фіксована як “+”.

“GM\_Orientable Surface” подають елементом об’єкту “OrientableSurface” у GML. Роль “примітив” подають властивістю “baseSurface”.

Для уможливлення використання “CompositeCurve” у GML, де взагалі очікують геометричний примітив, реалізовано абстрактний (та без властивостей) елемент об’єкту “AbstractCurve”, який можна замінити на або “Curve”, або “OrientableCurve”, або “CompositeCurve”. Такий же механізм використовують з поверхнями та тілами. Як результат, клас “GM\_OrientablePrimitive” не перекладають у GML явним чином, однак, оскільки цей тип не має екземплярів, це не накладає жодних обмежень.

**Примітка 1.** Цей перехід є наслідком того факту, що у просторовій схемі використане множинне успадкування для зазначення того, що композитно геометрією, яка за визначенням є комплексною геометрією, можна також подавати геометричний примітив. Оскільки XML схема нездатна на множинне успадкування, (або точніше: походження з множинних типів), елементи

абстрактного об'єкту “AbstractCurve”, “AbstractSurface” та “AbstractSolid” реалізовані у GML, щоб дозволити і “істинним” геометричним примітивам (наприклад, “Curve”), і композитним геометріям (наприклад, “CompositeCurve”) бути у спільній групі заміни, хоча вони є структурно різними.



**Рисунок Г.7** – Крива

Як розглянуто вище, “GM\_Curve” подають елементом об’єкту “Curve” у GML. Роль “segment” (“відрізок”) подають як властивість масиву “segments” у GML.



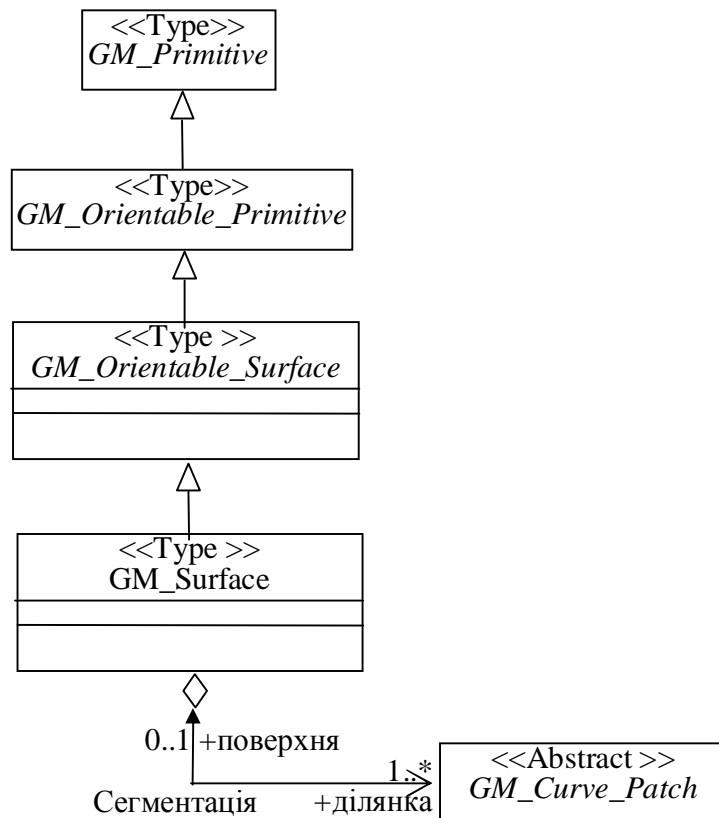


Рисунок Г.8 – Поверхня

Як розглянуто вище, “GM\_Surface” подають елементом об’єкту “Surface” у GML. Роль “patch” (“ділянка”) подають як властивість масиву “patches” у GML.

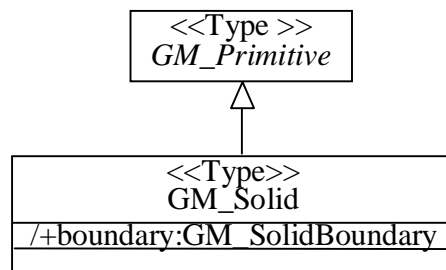


Рисунок Г.9 – Тіло

“GM\_Solid” подають елементом об’єкту “Solid” у GML. Межу тіла Solid безпосередньо подають властивостями “exterior” та “interior” тіла, як розглянуто вище.

#### Г.2.3.4 Координатна геометрія

На діаграмах UML класів на рисунках Г.10-Г.19 показано профіль пакету координатної геометрії (доречно порівняти з ISO 19107:2003, рисунки 14- 21).

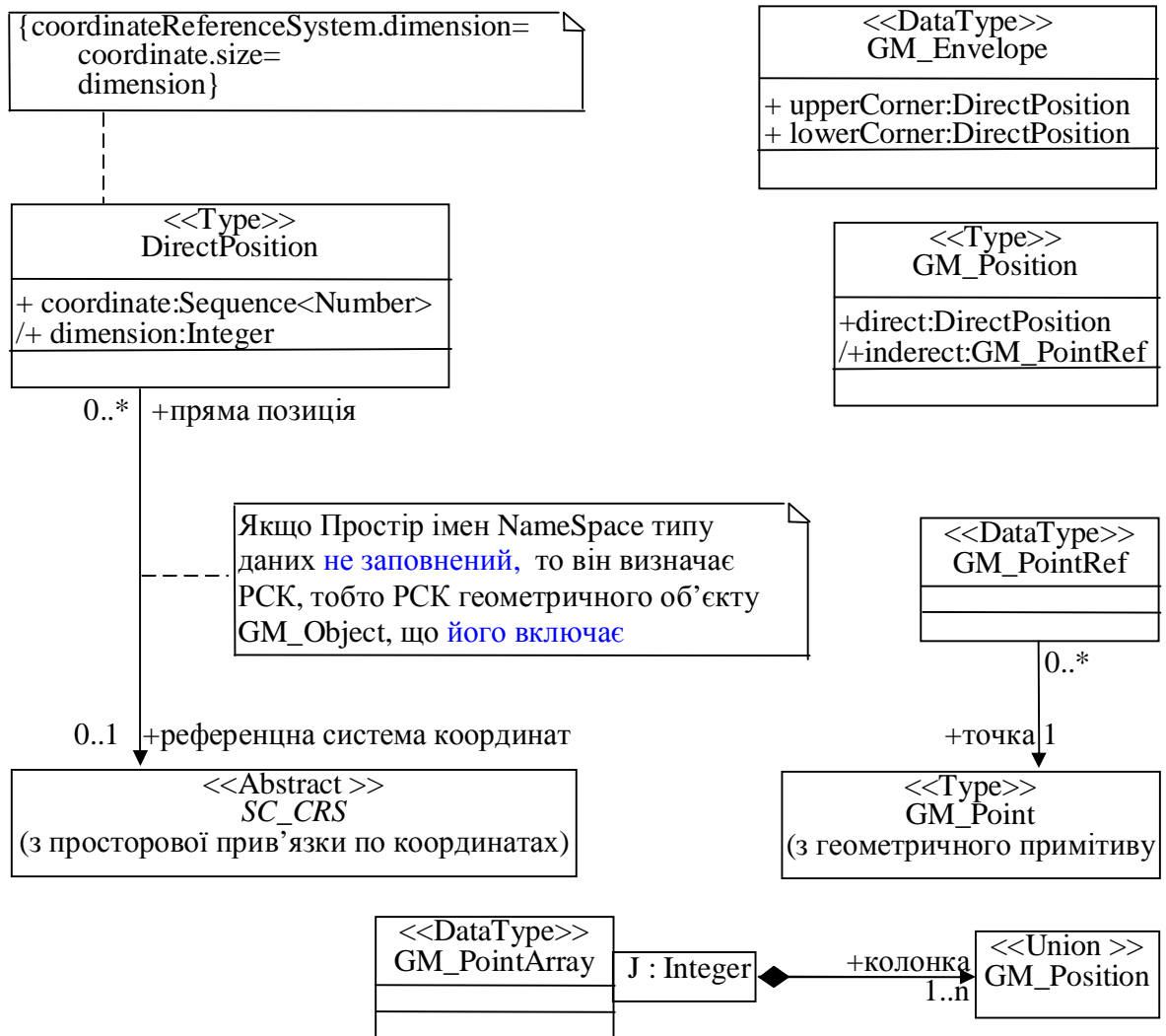


Рисунок Г.10 – Координатна геометрія

“DirectPosition” подають у GML як тип з простим контентом, де атрибут “coordinate” перекладено у список довгих десяткових чисел. Роль “референцна система координат” подають властивістю атрибуту “srsName”, та “dimension” подають необов'язковою властивістю атрибуту з таким же іменем (типом є додатне ціле число positiveInteger).

“GM\_Position” перекладають у вибір між властивістю “pos” (типом якої є “DirectPosition”) та властивістю “pointProperty” (значенням якої є “Point”). “GM\_PointArray” подають як подібний елемент вибору, але з відповідними установками для мінімального та максимального числа випадків).

Одиночні “GM\_Position” або “GM\_PointArray” можна альтернативно подати властивістю “coordinates” (типом значення є “Coordinates”, що є типом із простим контентом, яким подають список координат, закодований як текстовий рядок).

“GM\_Envelope” подають елементом об'єкту “Envelope” у GML. Два атрибути “upperCorner” та “lowerCorner” перекладають у властивості з таким же іменем. Додатковий атрибут “SRSReferenceGroup” у “gml:Envelope” додано для того, щоб

референцна система координат вимагала визначення тільки один раз у типовому випадку кутів в одній і тій же системі координат.

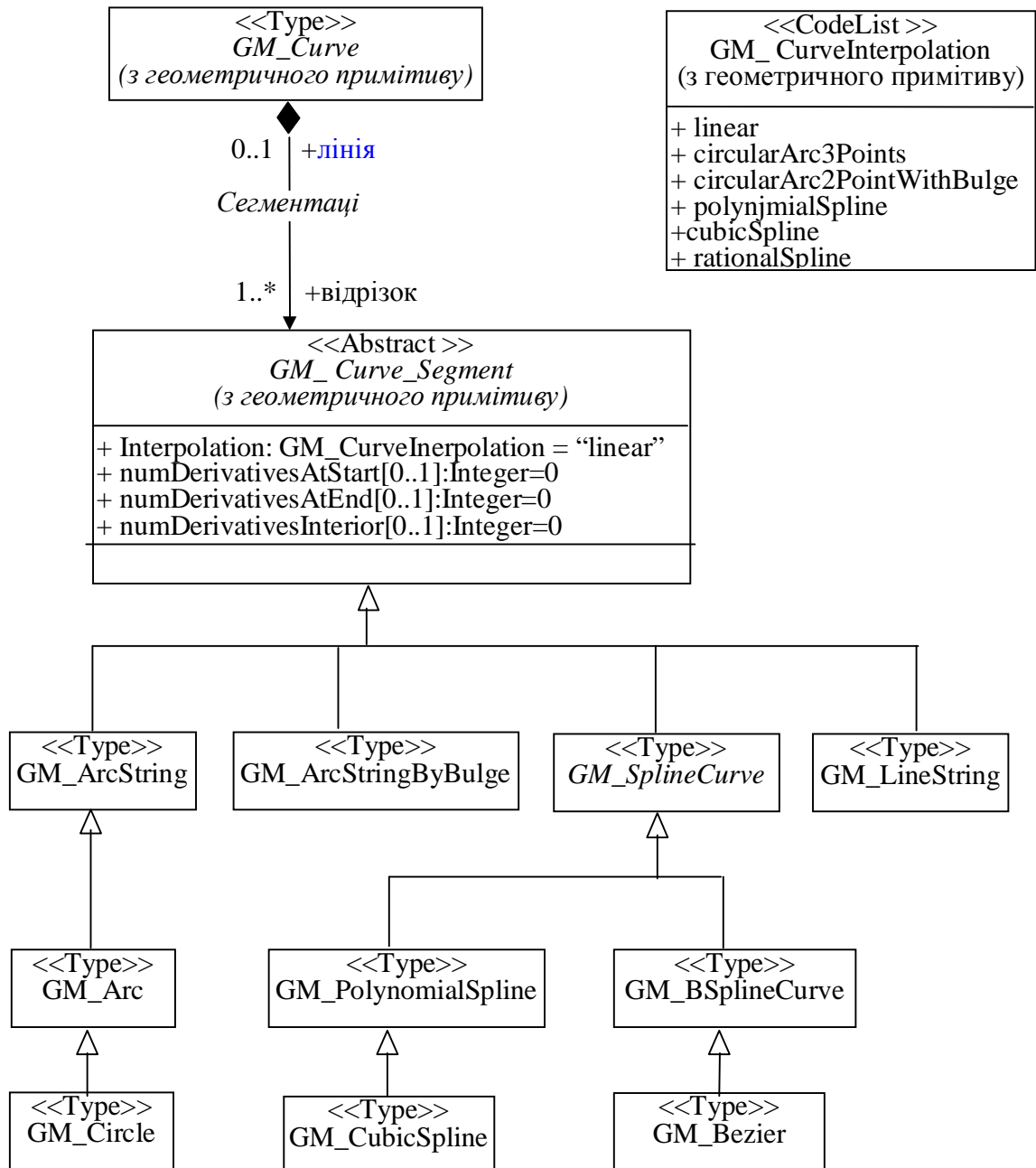


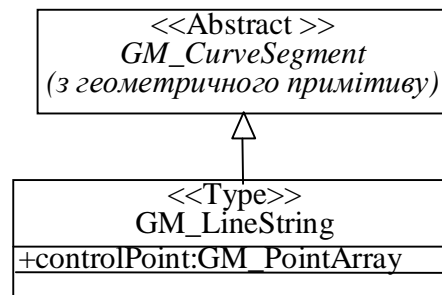
Рисунок Г.11 – Сегменти кривої

“GM\_CurveSegment” поданий у GML елементом об’єкту “AbstractCurveSegment” (обидва абстрактні). Три атрибути “numDerivatives...” переходять у властивості з таким же визначенням. Атрибут “interpolation” не визначено у “AbstractCurveSegment”, але визначено (та встановлено у відповідних початкових значеннях) у підтипах, що можуть мати екземпляри.

GML зараз підтримує піднабір визначених відрізків кривої стандарту ISO 19107.

Більшість підтипів для “GM\_CurveSegment” мають атрибут “controlPoint”, що подано у GML елементом вибору, як описано вище (див. розгляд подання GM\_PointArray).

Список кодів “GM\_CurveInterpolation” перекладено у GML так, неначе б він був переліком, тобто жодні додаткові значення не дозволені в додаток до наперед визначених значень у GML схемі.



**Рисунок Г.12** – Лінійний рядок

“GM\_LineString” подають елементом об’єкту “LineStringSegment”. Суфікс “Segment” додають до імені у GML, оскільки ім’я “LineString” уже зарезервовано для іншого елементу об’єкту в GML (див. Г.3.5). Для підтримки зворотної сумісності з попередніми версіями GML було неможливим змінити ім’я існуючого елементу, оскільки якщо б попереднє використання “LineString” було скасоване, це ім’я не було б наявним для реалізації GM\_LineString.

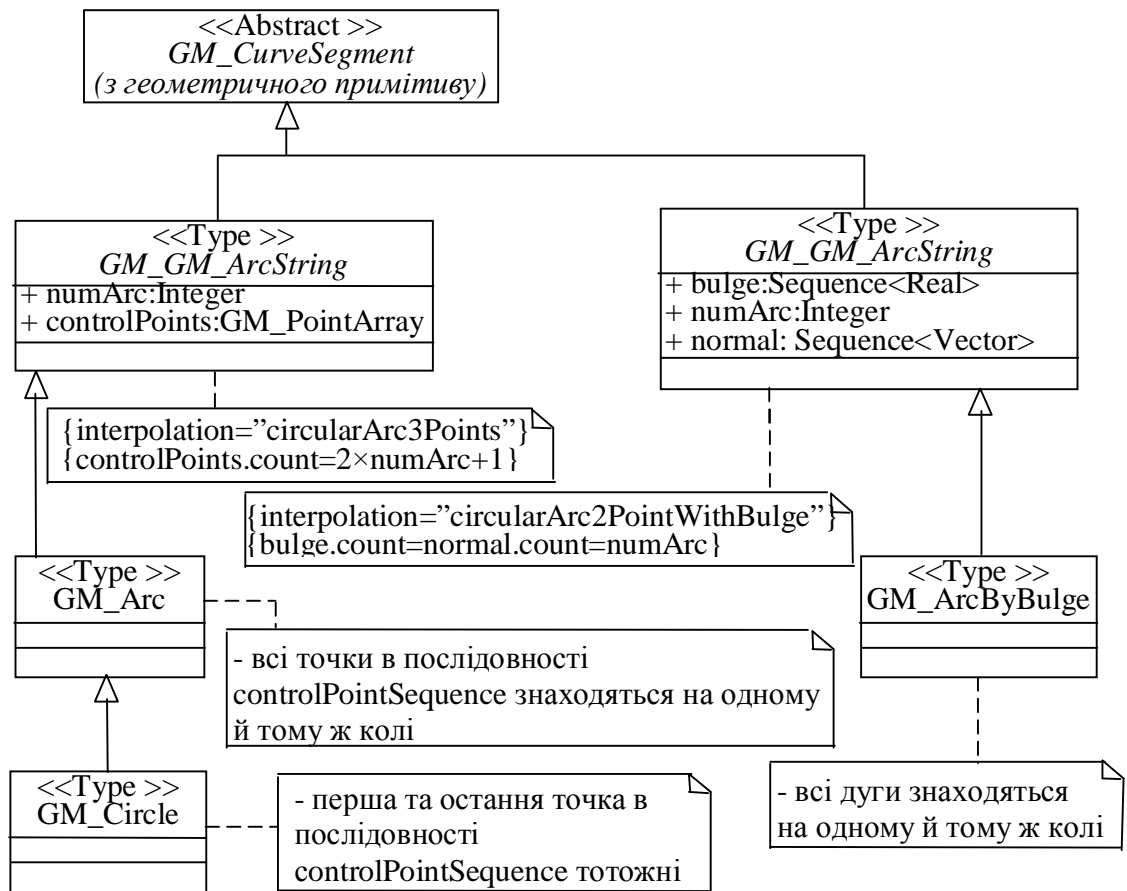


Рисунок Г.13 – Дуги та кола

Типи сегментів кривих переходять в елементи об'єктів у GML з тими ж іменем (але без префіксу "GM\_") та набором властивостей.

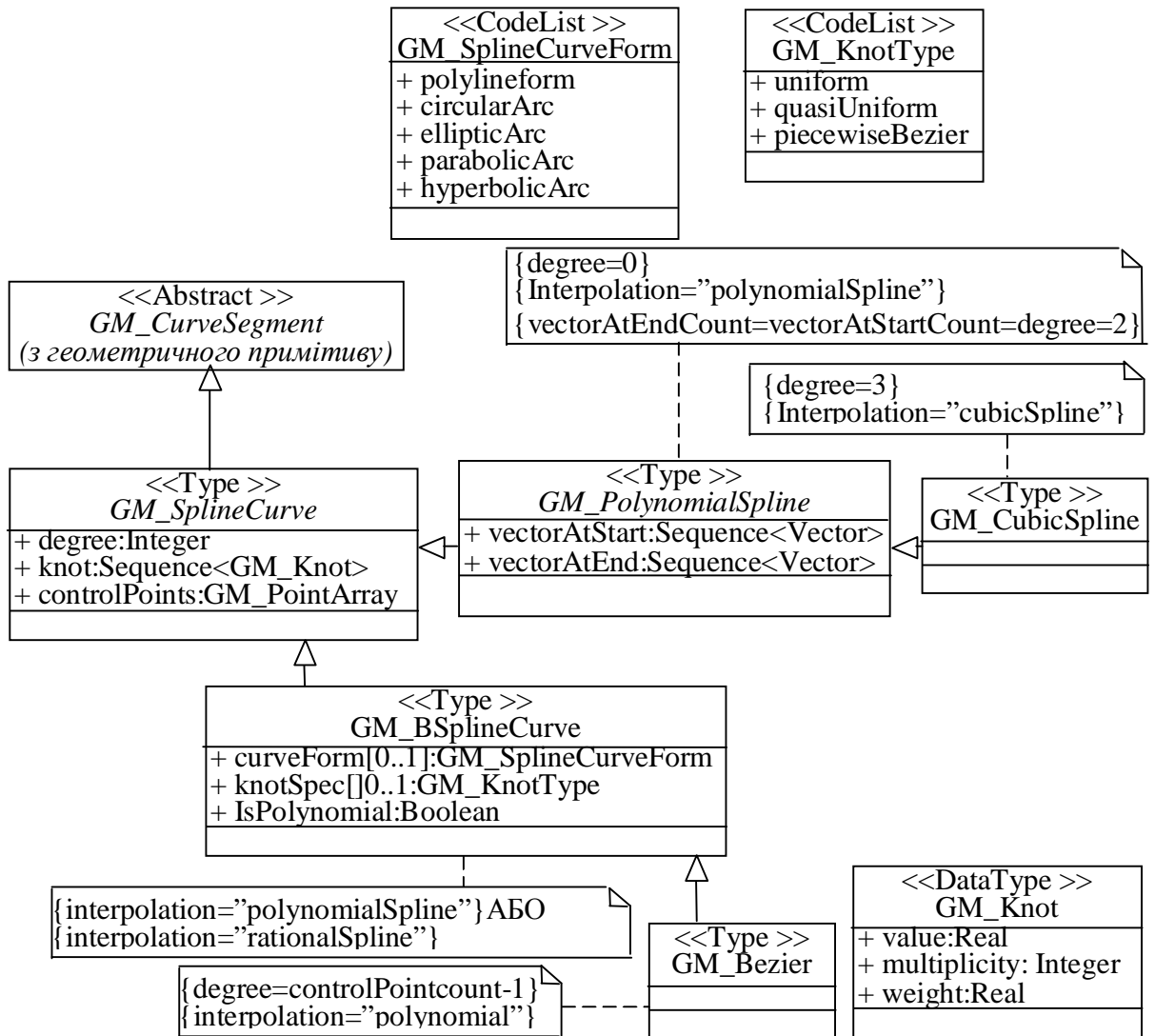


Рисунок Г.14 – Сплайни

Знову таки, ці типи відрізків кривих переходять в елементи об'єктів у GML із тими ж іменем, але без префіксу “GM\_”<sup>8)</sup> та тими ж властивостями<sup>9)</sup>. Властивості об'єктів сегментів кривої у GML визначені з урахуванням OCL обмежень.

Список кодів “GM\_KnotType” перекладений у GML, неначе б він був переліком, тобто жодні додаткові властивості не дозволені в додаток до наперед визначених значень у GML схемі.

<sup>8)</sup> Однак, “GM\_BSplineCurve” подають через “Bspline”, тобто без суфіксу “Curve”. Імя “Bspline” збережено для зворотної сумісності з попередніми версіями GML

<sup>9)</sup> Атрибут “knotSpec” було перейменовано на “knotType” у GML

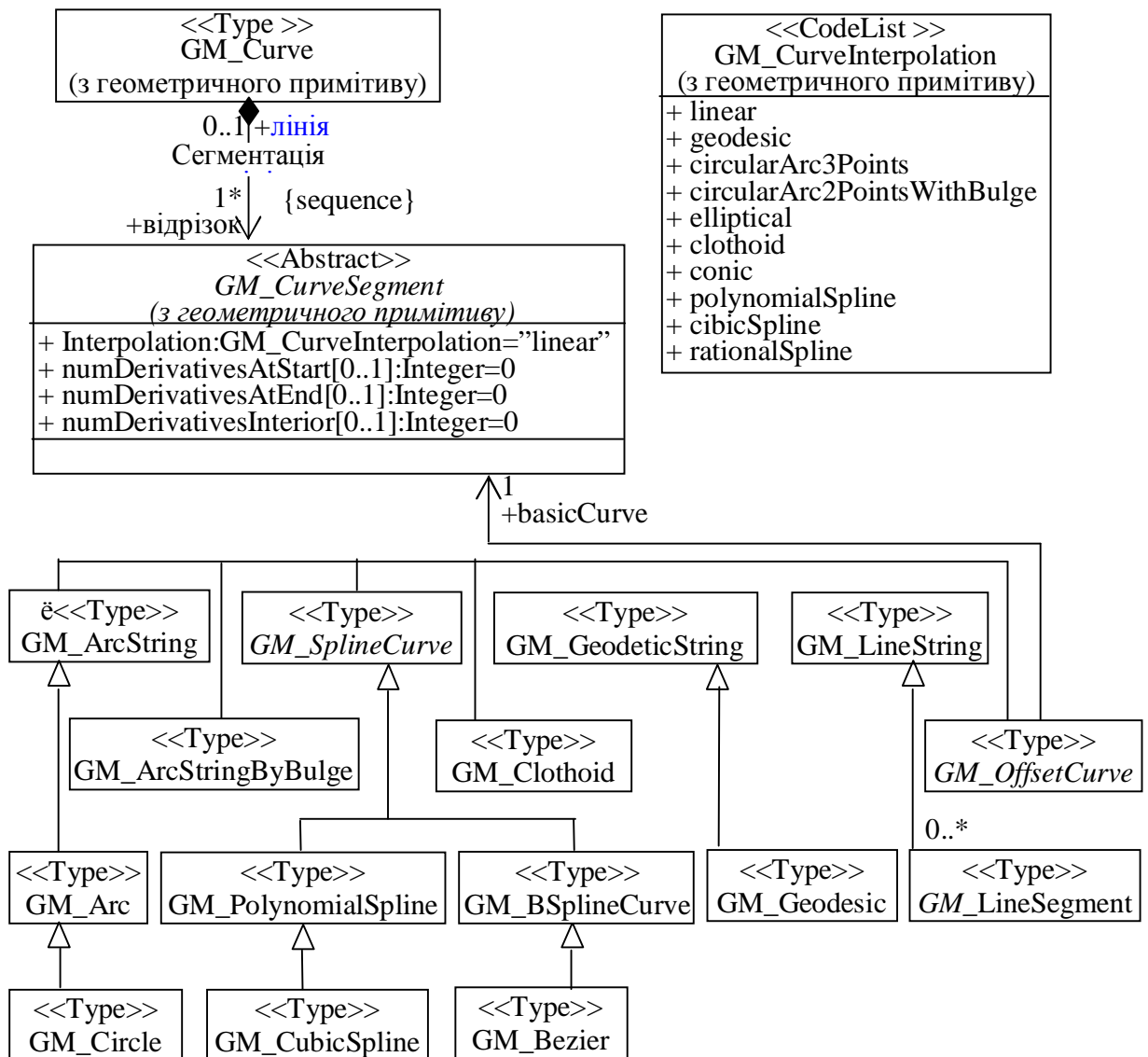


Рисунок Г.15 – Сегменти кривої

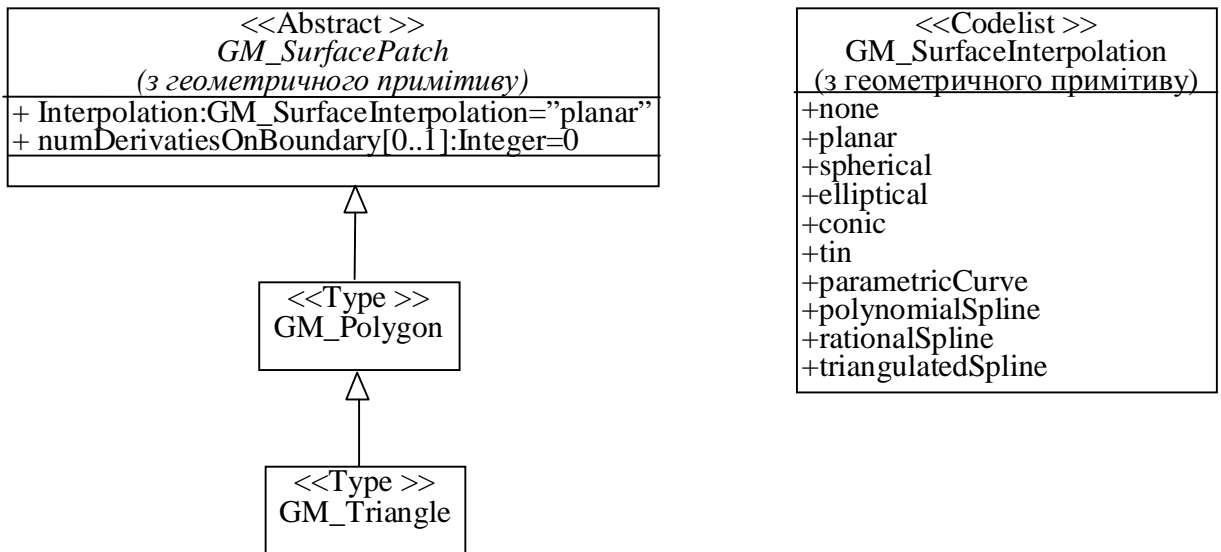
Клас “GM\_OffsetCurve” подають у GML елементом об’єкту “OffsetCurve”. Об’єкт має таку ж саму семантичну інтерпретацію, що цей клас. Властивість baseCurve property перейменована на offsetBase.

“GM\_AffinePlacement” подають у GML елементом об’єкту “AffinePlacement”.

“GM\_GeodesicString” подають у GML елементом об’єкту “GeodesicString”.

“GM\_Geodesic” подають у GML елементом об’єкту “Geodesic”.

“GM\_Clothoid” подають у GML елементом об’єкту “Clothoid”.

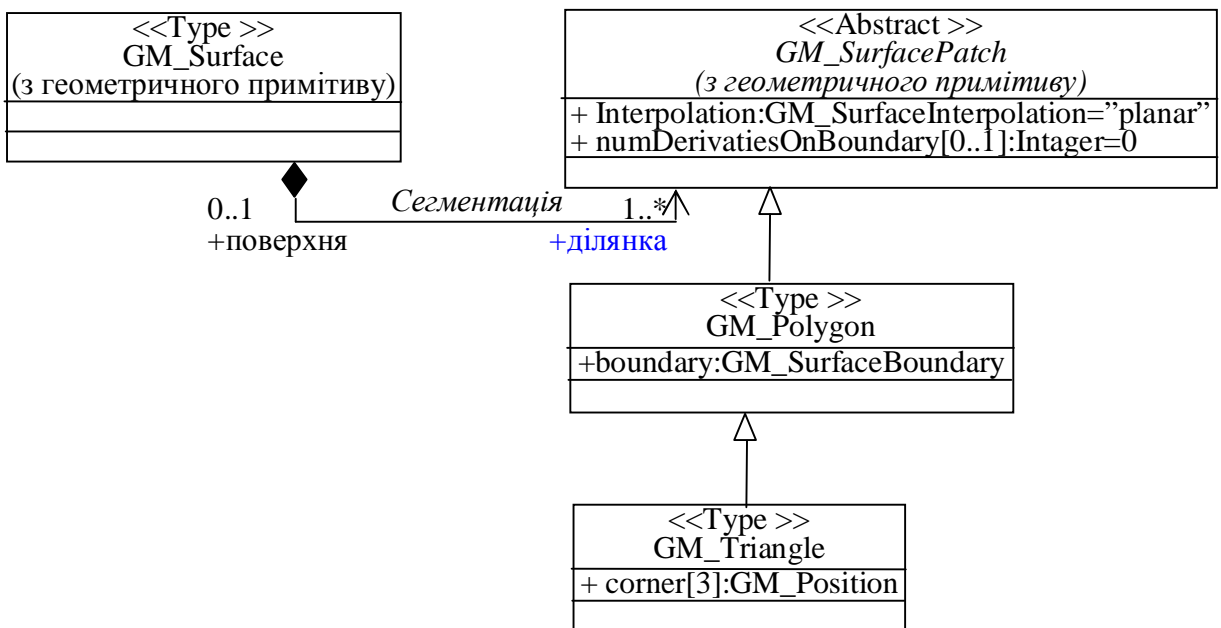


**Рисунок Г.16** – Ділянки поверхні (перший рисунок)

“GM\_SurfacePatch” подають у GML елементом об’єкту “AbstractSurfacePatch” (обидва є абстрактними). Атрибут “numDerivativesOnBoundary” зараз не перекладений явним чином у GML, оскільки в GML зараз підтримується тільки планарна інтерполяція. Атрибут “interpolation” не визначений у “AbstractSurfacePatch”, але він визначений (а також установлений із відповідними початковими значеннями) у підтипах, що мають екземпляри.

GML зараз підтримує піднабір всіх визначених типів поверхні та типів ділянок поверхні стандарту ISO 19107.

Список кодів “GM\_SurfaceInterpolation” був перекладений у GML, неначе б він був би переліком, тобто не дозволені жодні додаткові значення, крім тих, що були наперед визначені у GML схемі.

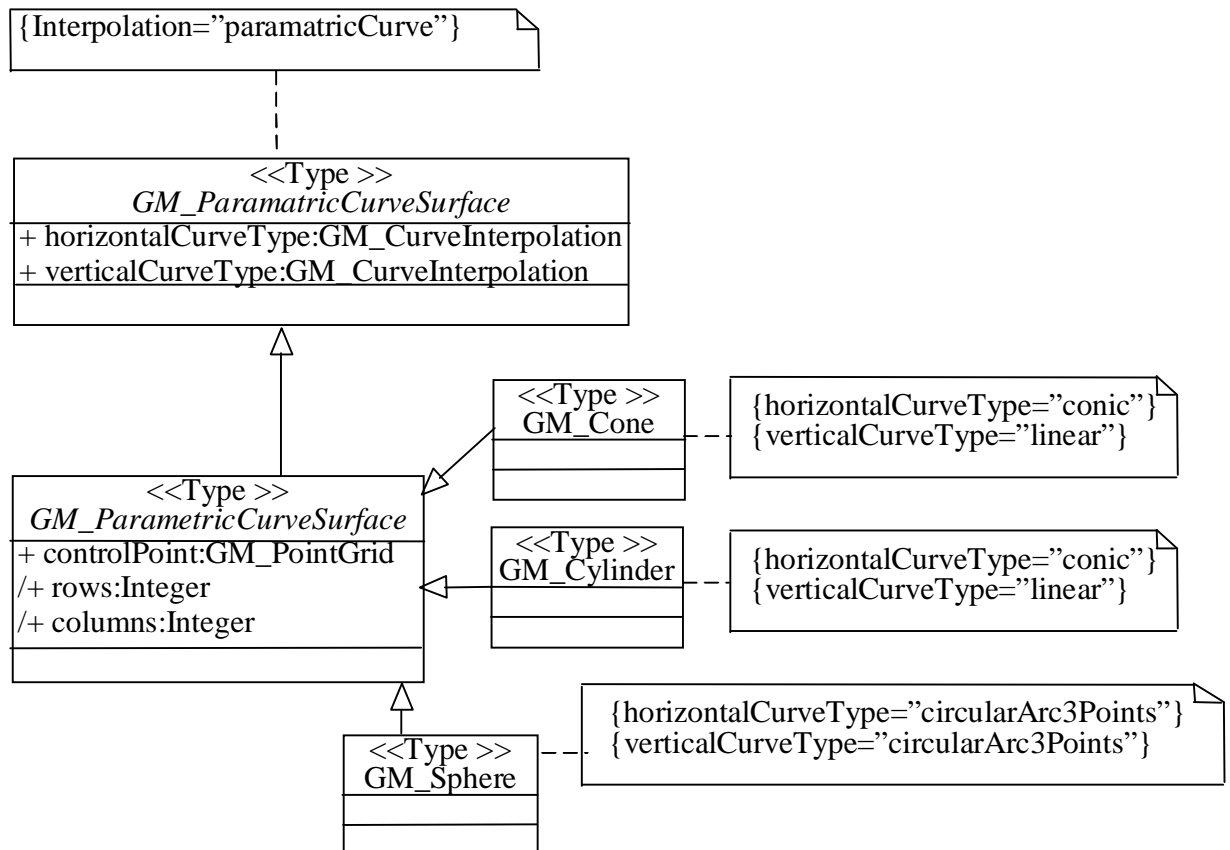


**Рисунок Г.17** – Ділянки поверхні (другий рисунок)



“GM\_Polygon” подають елементом об’єкту “PolygonPatch”. Атрибут “boundary” безпосередньо виражений властивостями “exterior” та “interior”, які належать до “PolygonPatch”.

Суфікс “Patch” був приєднаний до імені в GML, тому що ім’я “Polygon” уже було зарезервоване для іншого елемента об’єкту в GML (див. Г.3.6). Для підтримки зворотної сумісності з попередніми версіями GML було неможливим змінити ім’я існуючого елемента, оскільки навіть якщо б попереднє використання “Polygon” було скасоване, це ім’я не було б наявним для реалізації GM\_Polygon.



**Рисунок Г.18** – Ділянки поверхні в формі регулярної мережі

“GM\_PointGrid” подають у GML групою “PointGrid”.

“GM\_ParametricCurveSurface” подають у GML елементом об’єкту “AbstractParametricCurveSurface” (обидва є абстрактними).

“GM\_GriddedSurface” подають у GML елементом об’єкту “AbstractGriddedSurface” (обидва є абстрактними).

“GM\_Cone” подають у GML елементом об’єкту “Cone”.

“GM\_Cylinder” подають у GML елементом об’єкту “Cylinder”.

“GM\_Sphere” подають у GML елементом об’єкту “Sphere”.

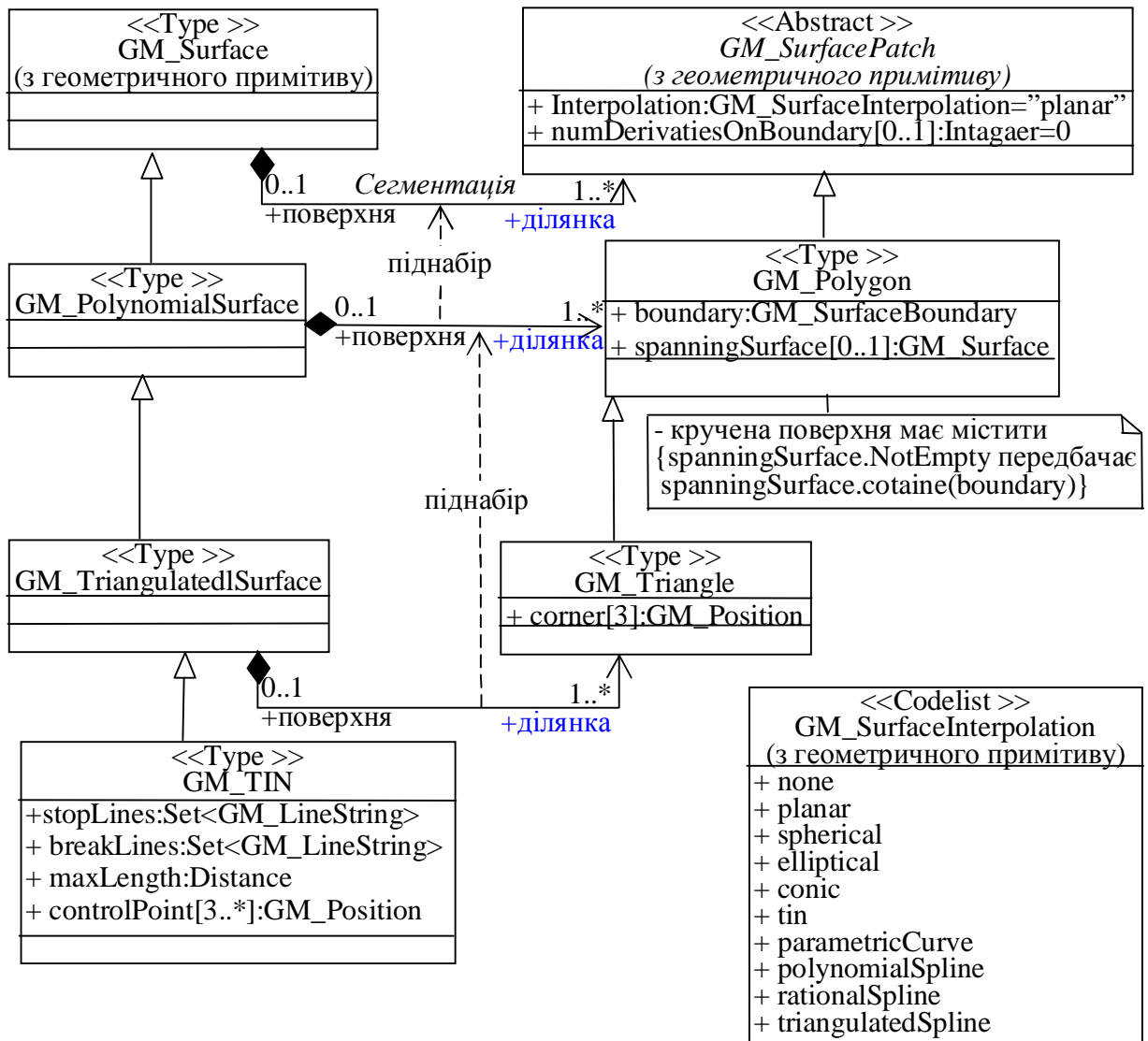


Рисунок Г.19 – Полігедральні та триангульовані поверхні

“GM\_PolyhedralSurface” подають у GML елементом об’єкту “PolyhedralSurface”.

“GM\_TriangulatedSurface” подають у GML елементом об’єкту “TriangulatedSurface”.

“GM\_Tin” подають у GML елементом об’єкту “Tin”.

### Г.2.3.5 Агрегати геометрії

На діаграмі UML-класу на рисунку Г.20 показано профіль пакету кореня геометрії “Geometry root” (доречно порівняти з ISO 19107:2003, рисунок 24).

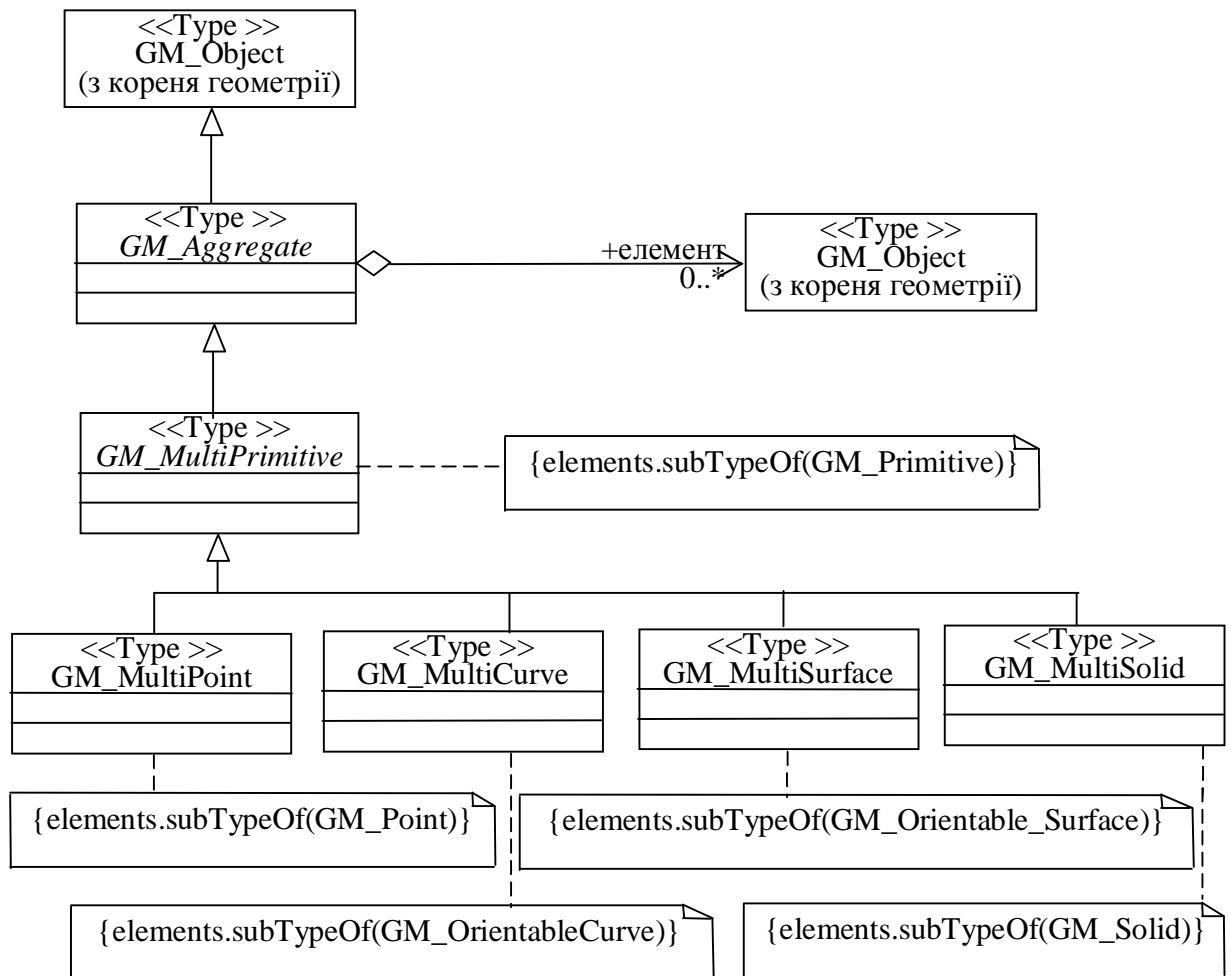


Рисунок Г.20 – Геометричні агрегати

“GM\_Aggregate” подають у GML елементом об’єкту “AbstractGeometricAggregate” (обидва є абстрактними). Роль “елемент” має екземпляри у GML у підтипах, які можуть мати екземпляри. Загальний шаблон полягає у тому, що визначені дві властивості, з них одна є регулярною властивістю асоціації, а інша є властивістю асоціації масиву. Іменами властивостей є “xMember” та “xMembers”, де відповідно “x” замінений на “point”, “curve”, “surface” або “solid” залежно від елементів колекції. Яким подають OCL обмеження для типу safetyess.

“GM\_MultiPoint” подають у GML елементом об’єкту “MultiPoint”.

“GM\_MultiCurve” подають у GML елементом об’єкту “MultiCurve”.

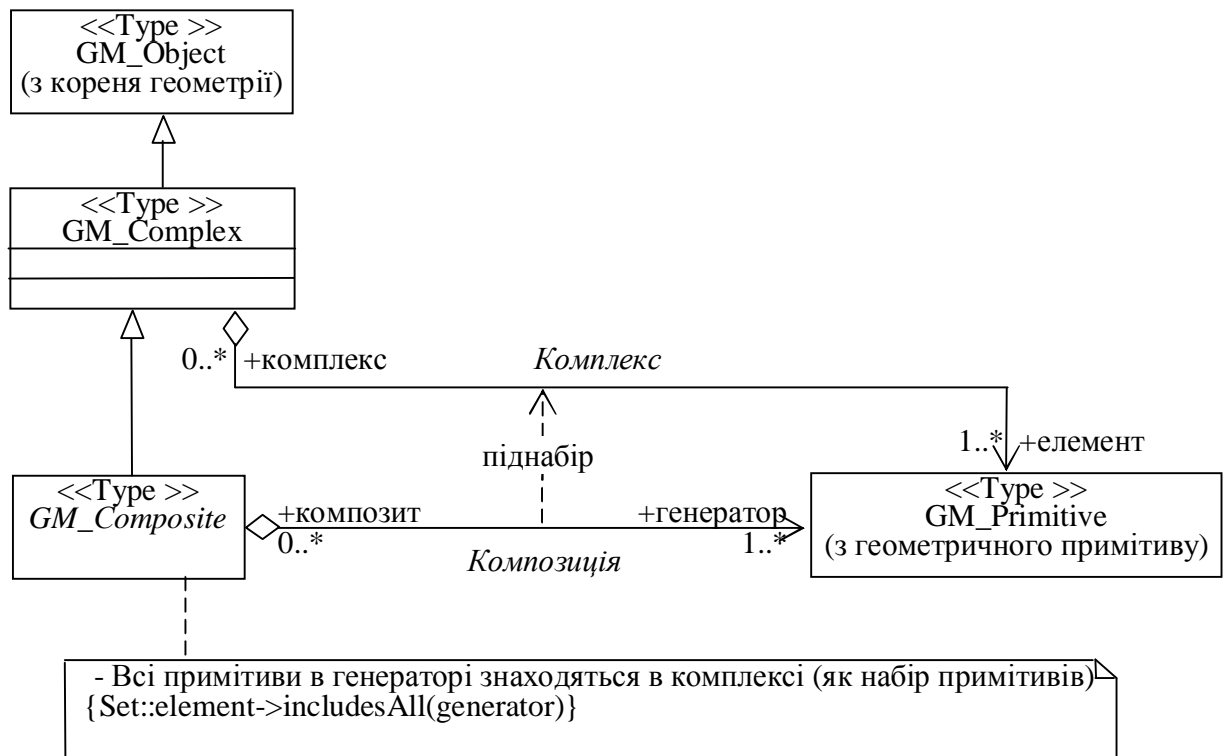
“GM\_MultiSurface” подають у GML елементом об’єкту “MultiSurface”.

“GM\_MultiSolid” подають у GML елементом об’єкту “MultiSolid”.

“GM\_MultiPrimitive” не має явного подання у GML.

### Г.2.3.6 Комплекс геометрії

На діаграмах UML класів на рисунках Г.21 та Г.22 відображено профіль пакету кореня геометрії “Geometry complex” (можна порівняти з ISO 19107:2003, рисунки 25-30).



**Рисунок Г.21** – Геометричні композити (перший рисунок)

“GM\_Complex” подано у GML елементом об’єкту “GeometricComplex”. Роль “елемент” переходить у властивість із тим же іменем у GML.

Той факт, що типи композитної геометрії є підтипами “GM\_Complex”, подають у GML так, що будь-яка властивість асоціації, яку приймає “GeometricComplex”, приймає також один із композитів (завдяки елементу вибору в “GeometricComplexPropertyType”). Необхідність у спеціальному перекладі до XML схеми була через те, що множинне успадкування, використане в ISO 19107 для вираження “дуалізму” композитних геометрій, не підтримується механізмом отримання XML схеми.

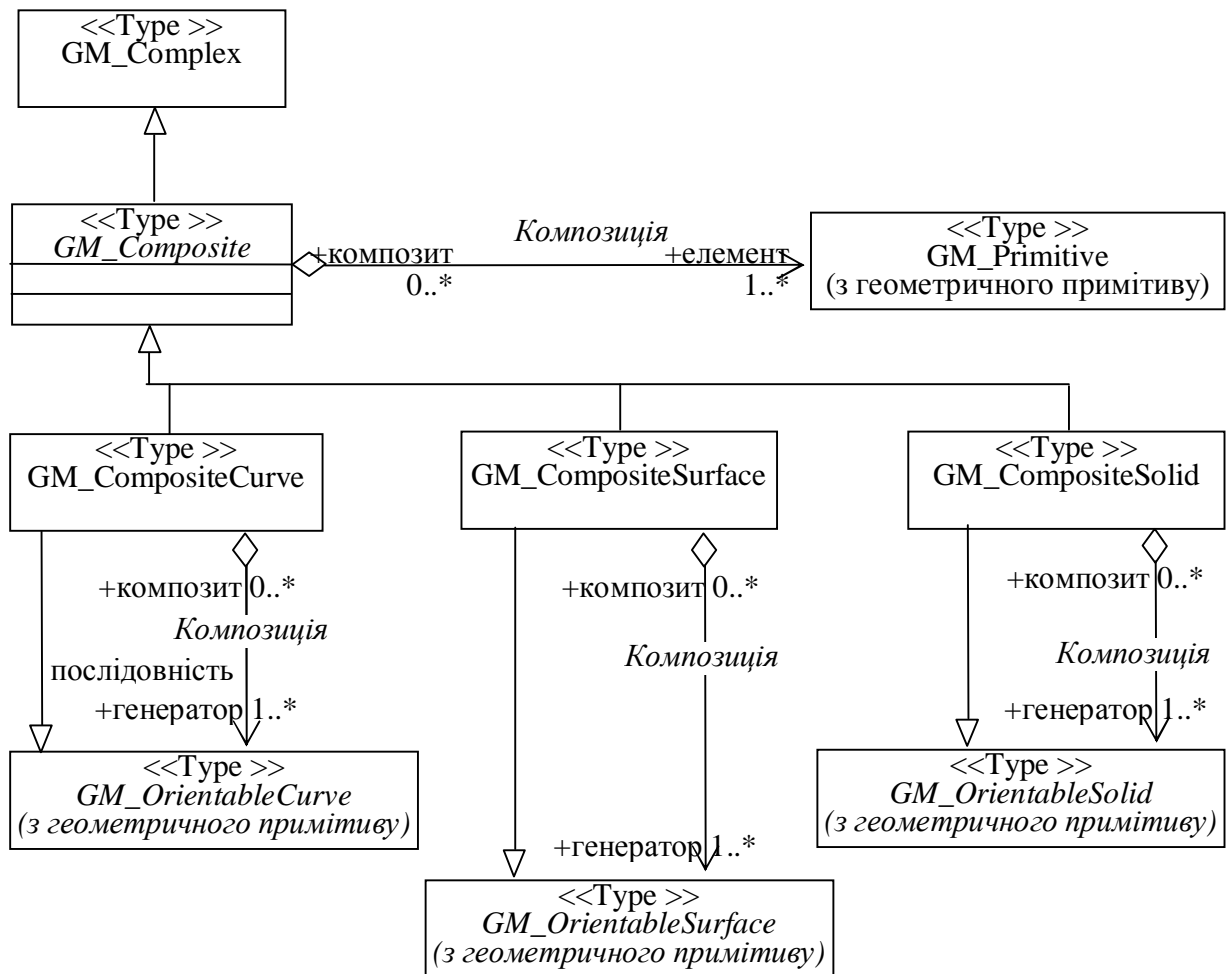


Рисунок Г.22 – Геометричні композити

“GM\_Composite” не подають явним чином як елемент об’єкту у GML. Однак, підтипи “GM\_CompositeCurve”, “GM\_CompositeSurface” та “GM\_CompositeSolid” подано у GML елементами об’єктів із тими ж самим іменем (без префіксу “GM\_”). Роль “generator” має екземпляри у GML у цих підтипах завдяки властивості асоціації з іменем “xMember”, де “x” замінений на “curve”, “surface” або “solid” залежно від елементів колекції.

### Г.2.3.7 Відповідність

Правила, які керують відповідністю профілю ISO 19107, описано в 19107:2003, розділ 2 та додаток А. Стосовно цих трьох критеріїв, визначених у розділі 2, геометрія GML охоплює такі рівні:

- а) комплексність даних;
  - геометричні примітиви;
  - геометричні комплекси;
- б) розмірність – нуль, одно-, двох- та трьохвимірні об’єкти;
- в) функціональна складність - тільки типи даних.

Таким чином, доречними підрозділами відповідності стандарту ISO 19107 є:

- А.1.1.1 - А.1.1.4

– А.2.1.1 - А.2.1.3.

Умови цих розділів відповідності виконані.

Треба зауважити, що похідні атрибути розглядають як операції і припускають, що похідні атрибути в агрегатних геометріях будуть отримані з даних з допомогою аплікації, яка працює з GML екземплярами.

GM\_CompositePoint подають елементом об'єкту "Point" у GML. Значенням ролі "генератор" в асоціації є той же об'єкт, тобто об'єкт "Point" як такий.

## Г.2.4 Просторова схема (топология) стандарту ISO 19107

### Г.2.4.1 Огляд

Додаткові зміни, показані в таблиці Г.4, застосовані до пакету топології стандарту ISO 19107.

**Таблиця Г.4** – Опис профілю стандарту ISO 19107 (топология)

Зміна	Пояснення
TP_Complex: асоціацію "isMaximal()" додано як похідний атрибут.	Інформація не є доступною засобами структур наперед визначених даних топологічного комплексу TP_Complex. Атрибут визначений як похідний атрибут, яким подають результат операції "isMaximal()", як визначено в ISO 19107
TP_Object змінено з класу інтерфейсу у клас типу (однак, без будь-яких властивостей), і відношення реалізації від TP_Primitive та TP_Complex до TP_Complex змінені на відношення генералізації	Підтримка TP_Object як кореня для різних топологічних підтипів робить яснішим перехід у GML
Необов'язкову асоціацію "реалізація" між TP_Complex та GM_Complex видалено	Реалізацію можна отримати з реалізації примітивів, що містяться в топологічному комплексі
Роль "maximalComplex" було видалено з TP_Primitive	Зараз не підтримується у GML
TP_Boundary та підтипи, як і TP_Ring та TP_Shell, видалено	Використовують тільки в операціях

### Г.2.4.2 Корінь топології

На діаграмі UML класів на рисунках Г.23 та Г.24 відображений профіль пакету кореня топології "Topology root" (доречно порівняти з ISO 19107:2003, рисунки 32 та 33).

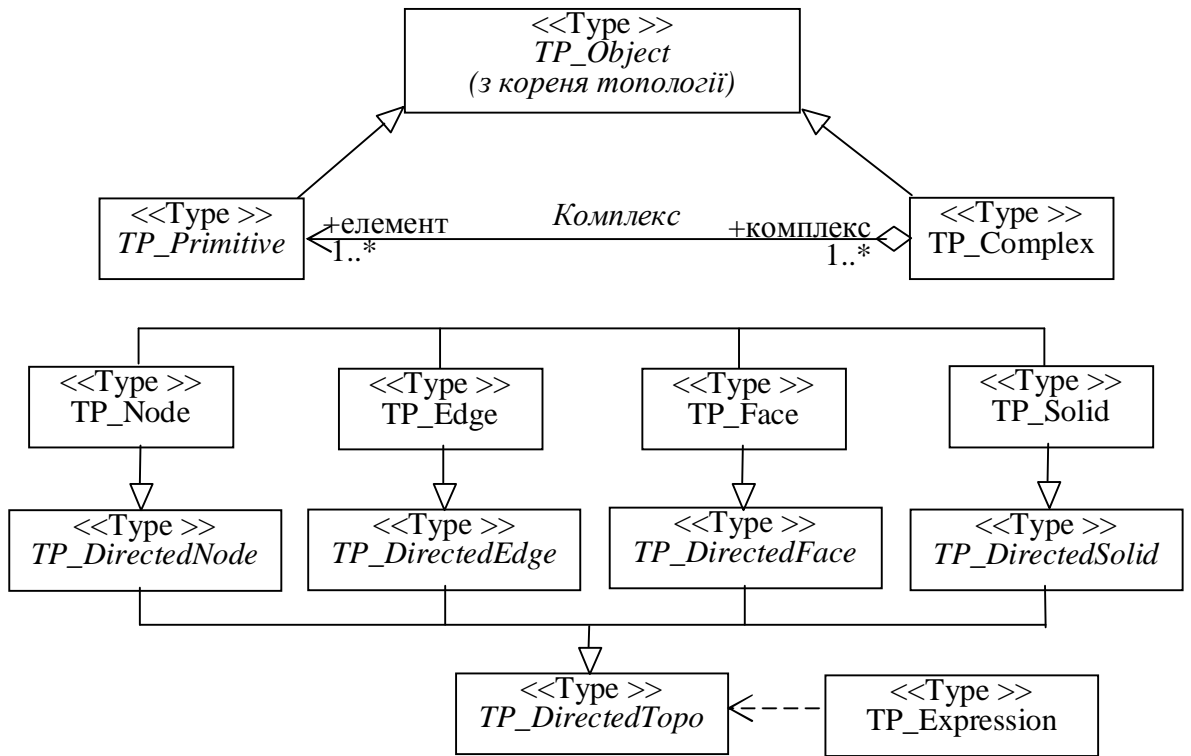


Рисунок Г.23 – Топологічні примітиви

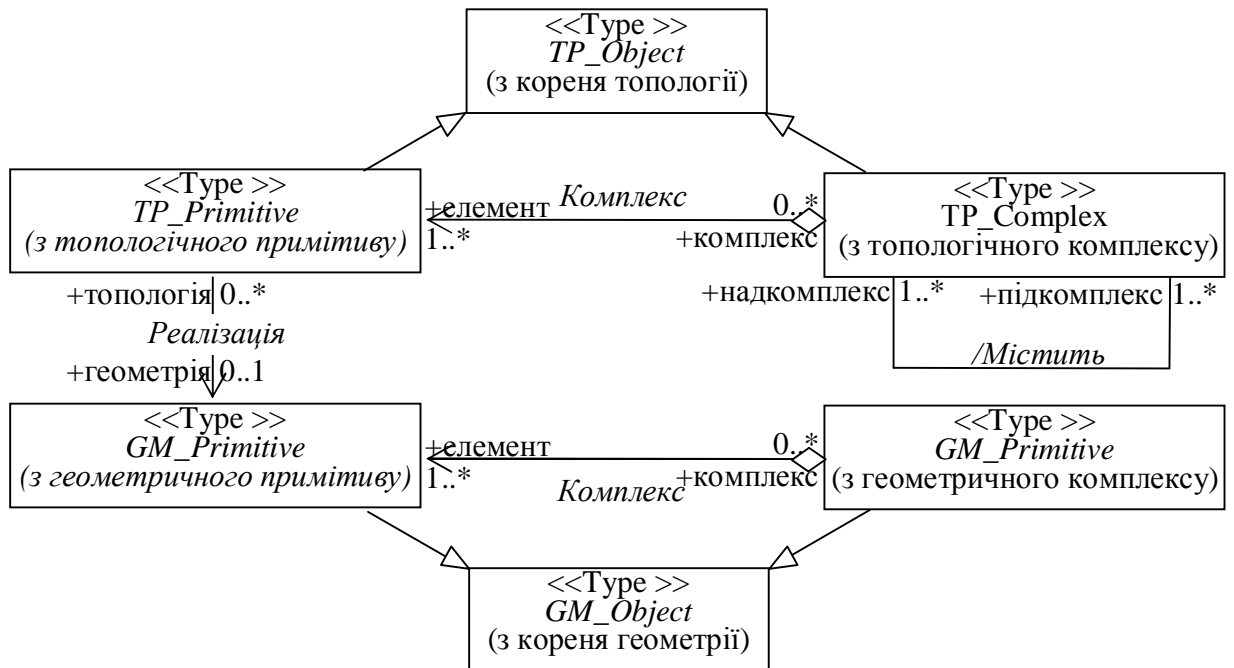


Рисунок Г.24 – Відношення між геометрією та топологією

Перехід різних класів у GML схему пояснений у наступних підрозділах, де показані деталі ієрархії класів.

“TP\_Object” поданий елементом об’єкту “AbstractTopology”. Елемент “AbstractTopology” може нести додаткові властивості: необов’язковий елемент “description”, необов’язковий елемент “identifier”, нуль чи більше елементів “name” та обов’язків атрибут “gml:id”.

#### Г.2.4.3 Примітив топології

На діаграмах UML класів на рисунках з Г.25 по Г.28 показано профіль пакету коріння топології “Topology root” (доречно порівняти з ISO 19107:2003, рисунки 35-45).

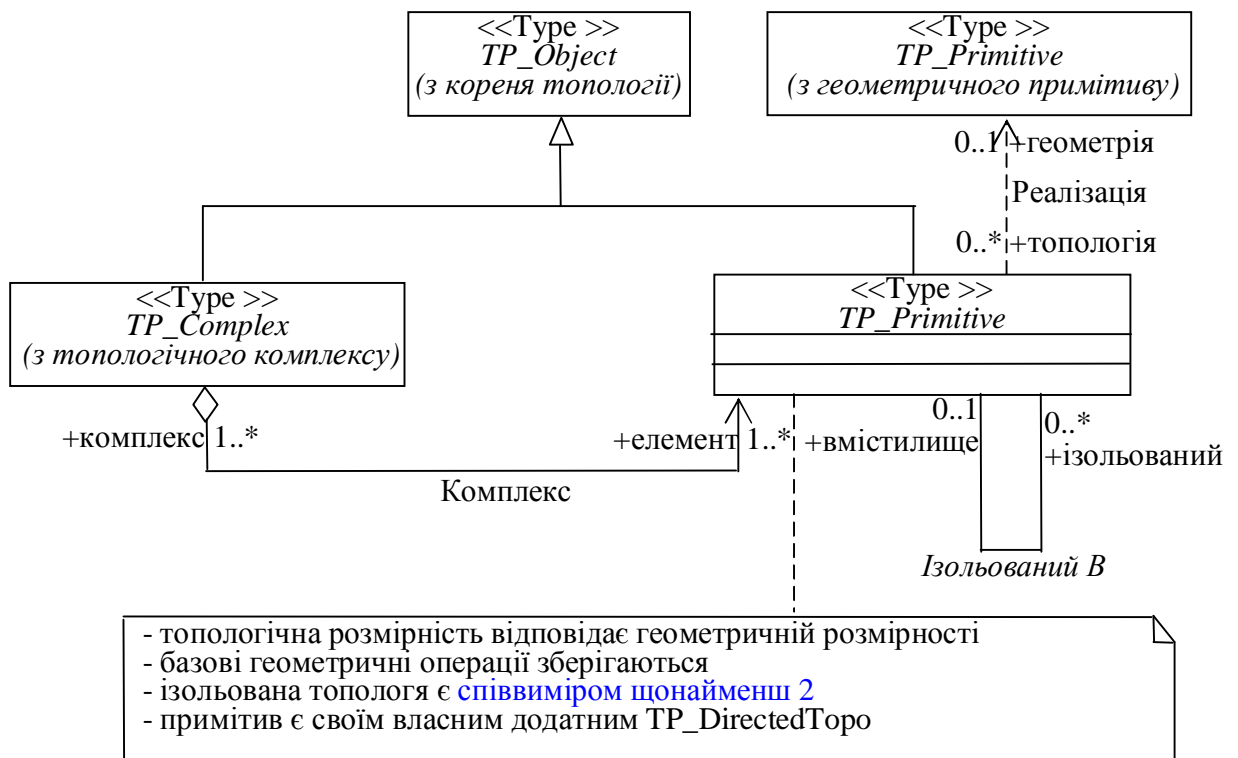


Рисунок Г.25 – Примітив топології

“TP\_Primitive” поданий елементом об’єкту “AbstractToroPrimitive”.

Роль “геометрія” (“geometry”) може мати екземпляри в підтипах, що можуть мати екземпляри. Це дозволяє контроль типів геометрії на іншому кінці асоціації (обмеження вимірності): “pointProperty”, “curveProperty”, “surfaceProperty” та “solidProperty” відповідно.

Ролі “ізолюваний” (“isolated”) та “вмістилище” (“container”) подають як властивості в “AbstractToroPrimitive”.



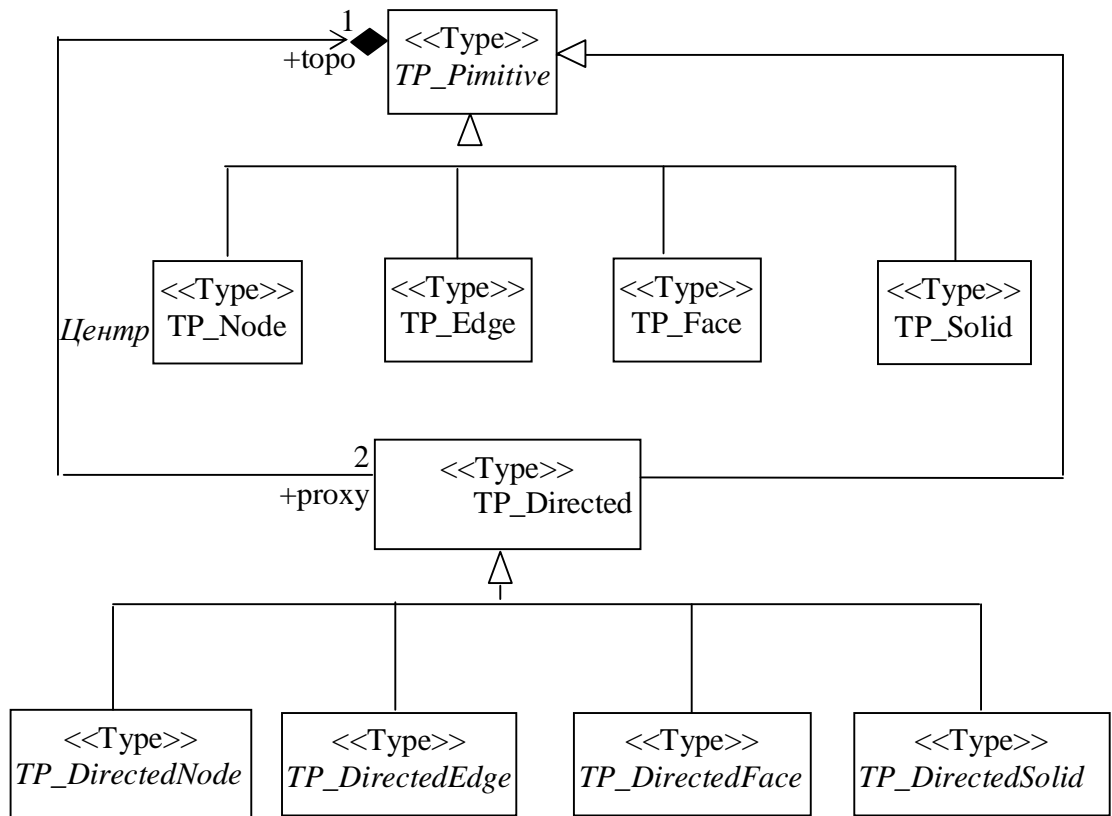


Рисунок Г.26 – Примітиви топології

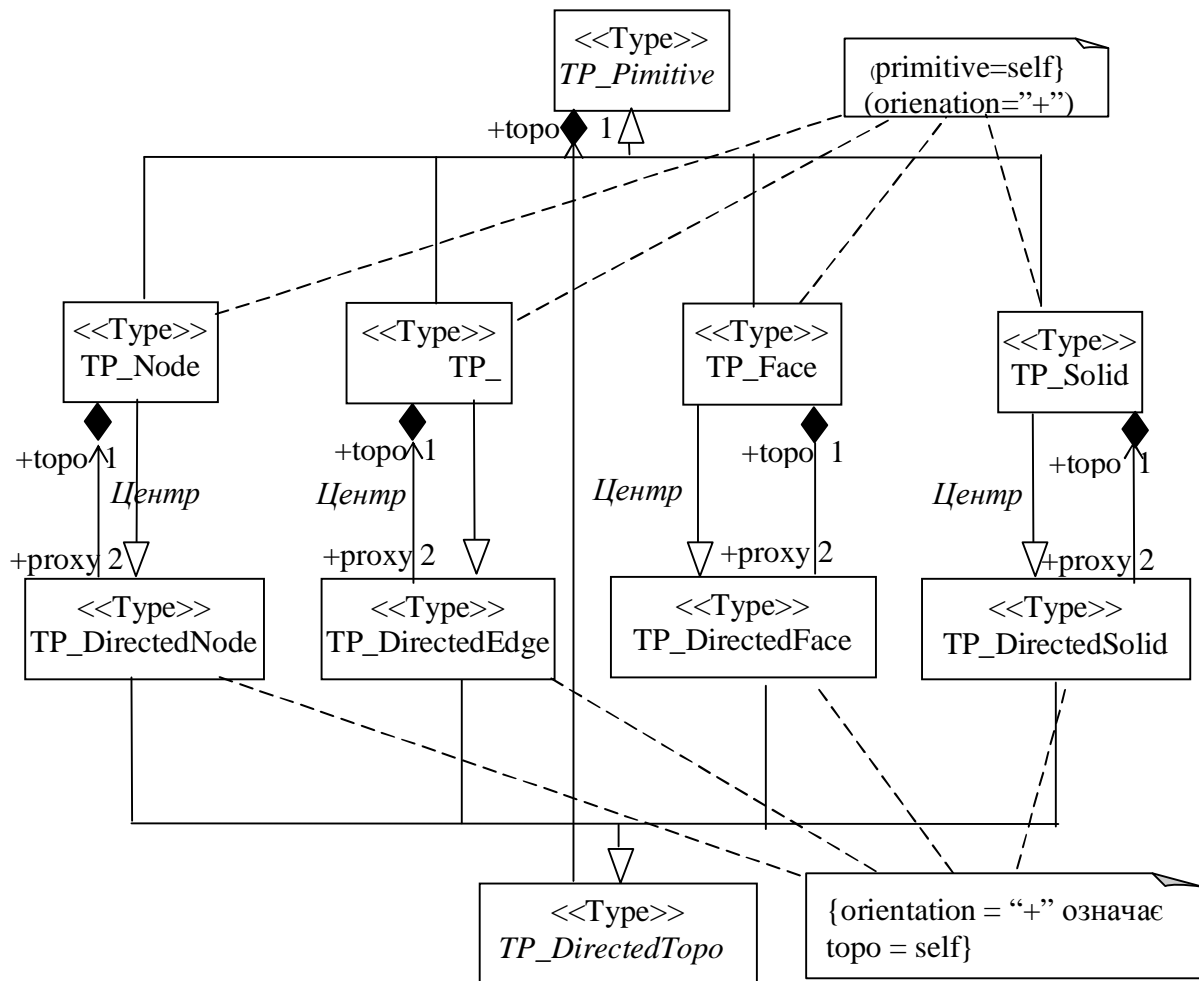


Рисунок Г.27 – Спрямовані примітиви топології

“TP\_Node” поданий елементом об’єкту “Node” у GML.

“TP\_Edge” поданий елементом об’єкту “Edge” у GML.

“TP\_Face” поданий елементом об’єкту “Face” у GML.

“TP\_Solid” поданий елементом об’єкту “TopoSolid” у GML (ім’я “Solid” вже використовують для трьохвимірному примітиву геометрії).

“TP\_DirectedTopo” не є явно поданим у GML, тільки в його типах, що можуть мати екземпляри. Помітною різницею є те, що, хоча типи спрямованої топології моделюють як типи, вони подані як властивості з атрибутом “orientation” у GML.

“TP\_DirectedNode” поданий як елемент властивості “directedNode” у GML. Роль “topo” подають безпосередньо елементом об’єкту “Node”.

“TP\_DirectedEdge” поданий як елемент властивості “directedEdge” у GML. Роль “topo” подають безпосередньо елементом об’єкту “Edge”.

“TP\_DirectedFace” поданий як елемент властивості “directedFace” у GML. Роль “topo” подають безпосередньо елементом об’єкту “Face”.

“TP\_DirectedSolid” поданий як елемент властивості “directedTopoSolid” у GML. Роль “topo” подають безпосередньо елементом об’єкту “TopoSolid”.

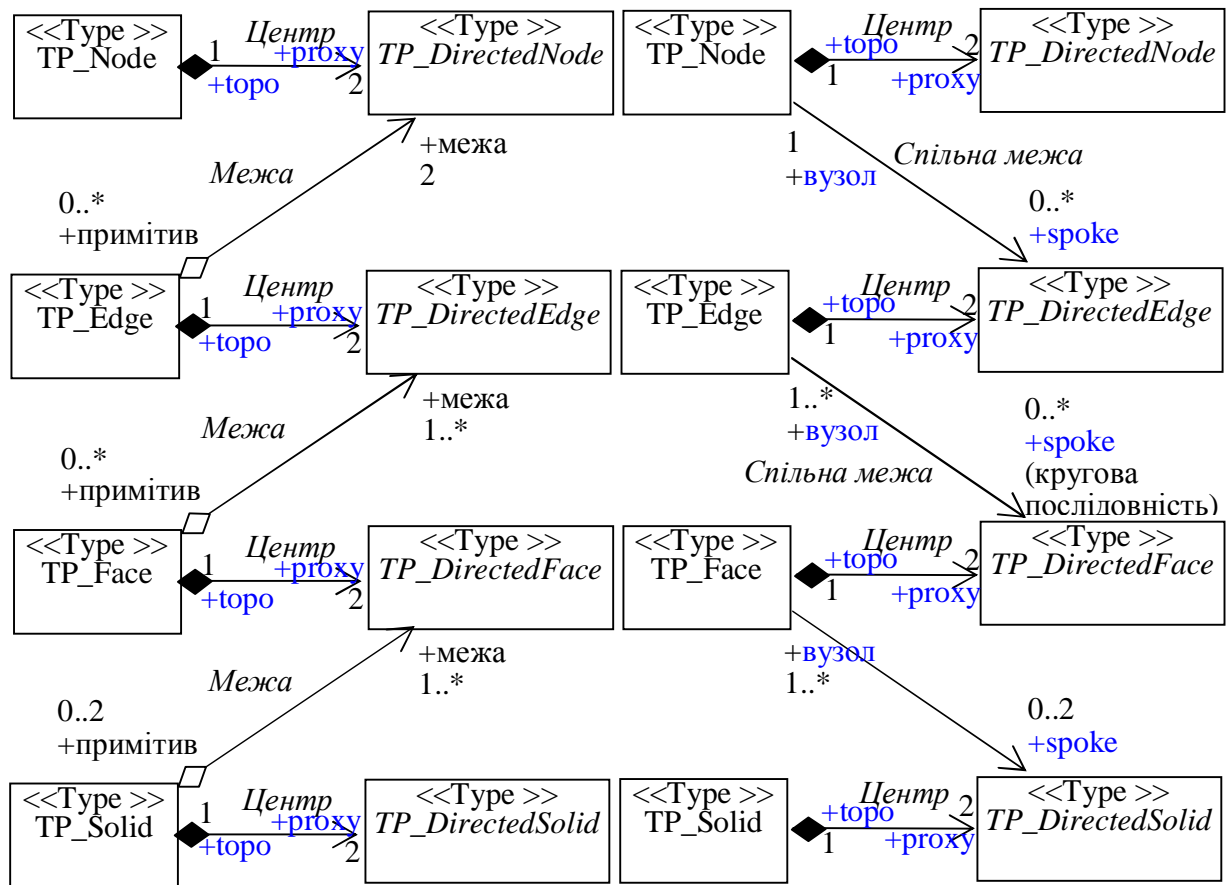


Рисунок Г.28 – Відношення меж та спільних меж

Перехід ролі “topo” вже розглянутий вище.

Роль “spoke” подана у GML властивостями “directed Edge”, “directedFace” та “directedTopoSolid” відповідно.

Ролі “boundary” (межа) подана у GML властивостями “directed Node”, “directedEdge” and “directedFace” відповідно.

#### Г.2.4.4 Комплекс топології

На діаграмі UML класів на рисунку Г.29 показано профіль пакету комплексу топології “Topology complex” (доречно порівняти з ISO 19107:2003, рисунок 46).

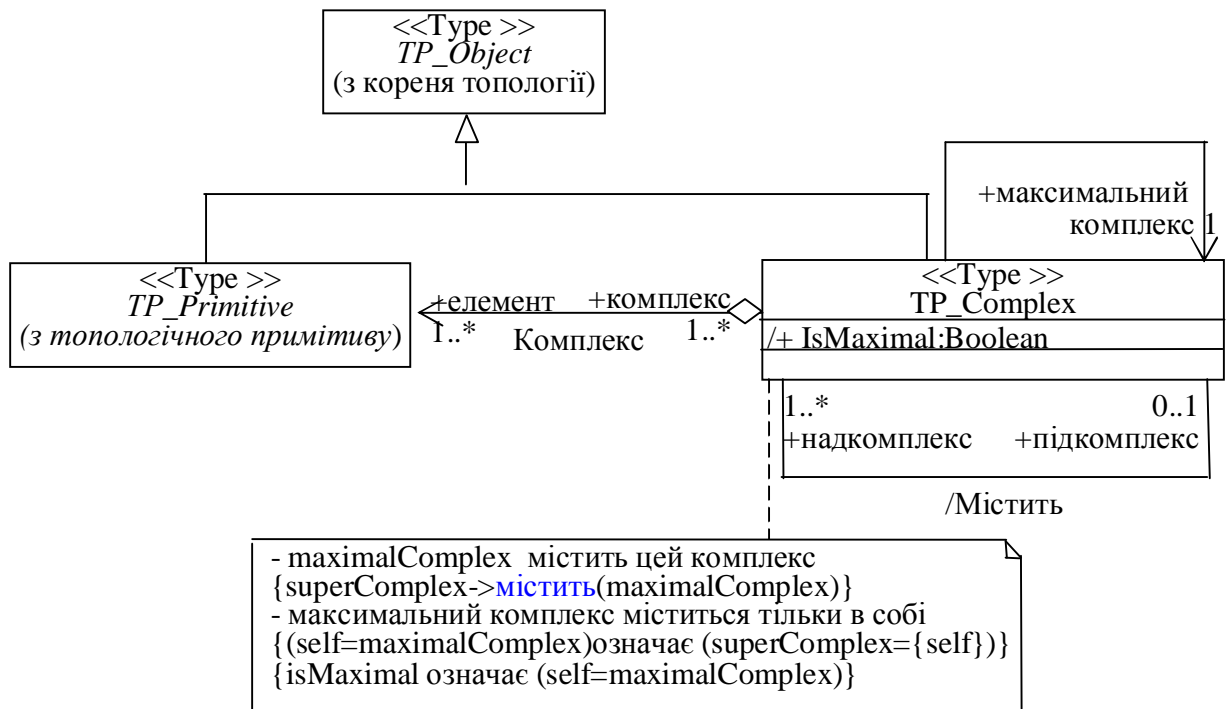


Рисунок Г.29 – Комплекс топології

“TP\_Complex” поданий елементом об’єкту “ТороComplex”.

Роль “елемент” (“element”) переходить у дві властивості: одна – властивість регулярної асоціації “topoPrimitiveMember” та друга – властивість масиву асоціацій “topoPrimitiveMembers”.

Ролі “підкомплекс” (“subComplex”) та “надкомплекс” (“superComplex”) та подані як властивості асоціацій того ж імені. Мінімальна чисельність, однак, є нулем для обох властивостей у GML замість 1. Це віддзеркалює відсутність необхідності явно подавати цю властивість в GML екземплярі (треба зауважити, що це є похідна асоціація).

Роль максимальний комплекс (“maximalComplex”) подана як властивість асоціації того ж імені в GML.

#### Г.2.4.5 Відповідність

Правила, що визначають відповідність профілю до ISO 19107, описані в ISO 19107:2003, розділ 2 та додаток А. Стосовно трьох критеріїв, визначених у розділі 2, GML топологія охоплює такі рівні:

- а) комплексність даних;
  - топологічні комплекси;
  - топологічні комплекси з геометричними реалізаціями;
- б) розмірність – нуль, одно-, двох та трьохвимірні об’єкти;
- в) функціональна складність - тільки типи даних.

Таким чином, доречними підрозділами відповідності стандарту ISO 19107 є:

- А.3.1.1 - А.3.1.3;
- А.4.1.1 - А.4.1.3.

Умови цих розділів відповідності виконані.

Треба зауважити, що асоціація “Realization” між TP\_Complex та GM\_Complex не є явною частиною профілю, тому що геометричну реалізацію топологічного комплексу можна отримати з геометричної реалізації топологічних примітивів.

## Г.2.5 Часова схема ISO 19108

### Г.2.5.1 Огляд

GML часові схеми забезпечують реалізацію стандарту ISO 19108:2002, підрозділи 5.2 - 5.4.

Зміни, показані в таблиці Table Г.5, були застосовані до пакетів ISO 19108.

**Таблиця Г.5** - Опис профілю стандарту ISO 19108

	<b>Зміна</b>	<b>Пояснення</b>
1.	Асоціації <i>beginning</i> та <i>ending</i> реалізовані як односпрямовані від TM_Period до TM_Instant	Забезпечення двоспрямованих покажчиків вважають за таке, що не є необхідним у кодуванні статичних даних
2.	Асоціації <i>beginning</i> та <i>ending</i> від TM_Period реалізовані як вибір (UML асоціація з TM_Instant або UML атрибутом типу TM_Position)	Дозволяє більш компактне кодування тієї ж самої інформації, коли об'єкт TM_Instant з ідентичністю не є необхідним
3.	Асоціації <i>termination</i> та <i>initiation</i> реалізовані як односпрямовані від TM_Edge до TM_Node	Забезпечення двоспрямованих покажчиків вважають за таке, що не є необхідним у кодуванні статичних даних
4.	Асоціації <i>Realization</i> реалізовані як односпрямовані від TM_Instant та TM_Period до TM_Node та TM_Edge відповідно	Забезпечення двоспрямованих покажчиків вважають за таке, що не є необхідним у кодуванні статичних даних
5.	Асоціація <i>basis</i> реалізована як односпрямована від TM_Calendar до TM_CalendarEra	Забезпечення двоспрямованих покажчиків вважають за таке, що не є необхідним у кодуванні статичних даних
6.	Атрибути <i>begin</i> та <i>end</i> для TM_OrdinalEra замінені асоціаціями з TM_Node, з іменами ролей <i>start</i> та <i>end</i>	Практика історичних та геологічних наук свідчить, що точки завершень порядкових ер асоційовані з подіями, позиції яких не можна точно визначити. Це відповідає концепції TM_Node, позиція якого не безпосередньо наявною
7.	Атрибут <i>origin</i> для TM_CoordinateSystem реалізований як вибір (UML асоціація з TM_Instant UML або атрибут типу TM_Position)	Забезпечує визначення початку в термінах зовнішньої події
8.	Атрибут <i>interval</i> для TM_CoordinateSystem реалізований як TM_Interval Length	Дозволяє більш точне та гнучке визначення масштабу

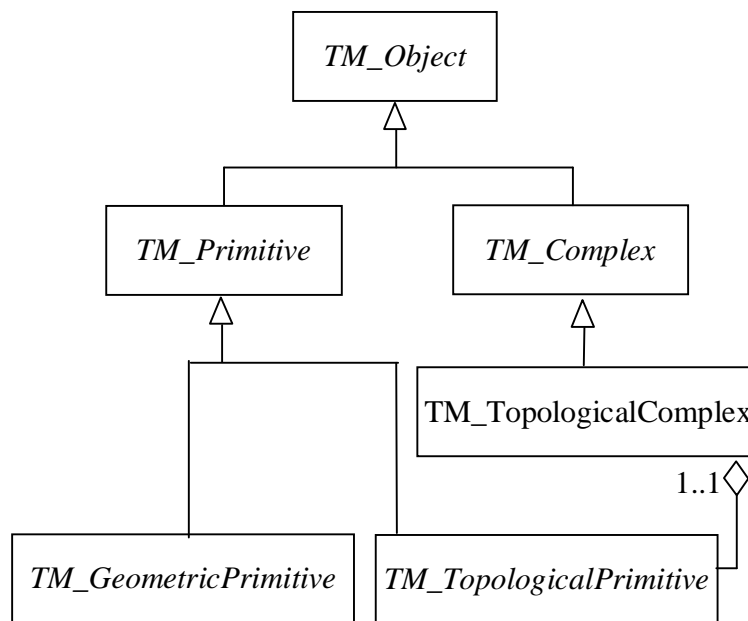
**Примітка 1.** Зміна 2 включає перевагу <choice> структури, яка забезпечена мовою реалізації XML схеми. Це підтримує більш компактне та гнучке кодування, що містить ту ж саму інформацію, ніж автоматичне застосування стандартних правил кодування.

Серед класів, які стосуються часових відносин між географічними об'єктами, описаних у підрозділі 5.5 стандарту ISO 19108:2002, була безпосередньо реалізована тільки Feature Succession (сукцесія географічних об'єктів). Компоненти, що відповідають іншим відношенням, можна визначити в GML прикладних схемах, але вони далі тут не розглянуті.

Перехід різних класів у GML схему пояснено у наступних підрозділах, де показано деталі ієрархії класів.

На UML діаграмах класів на рисунках від Г.30 до Г.34 показано профіль пакету часових об'єктів “Temporal Objects” (доречно порівняти з ISO 19108:2002, рисунки 2-6 та 11).

### Г.2.5.2 Часові об'єкти



**Рисунок Г.30** – Основна ієрархія часових об'єктів із стандарту ISO 19108

“TM\_Object” подають елементом об'єкту “AbstractTimeObject”. Елемент “AbstractTimeObject” може забезпечувати додаткові властивості: необов'язковий елемент “description”, нуль чи більше елементів “name”, необов'язковий елемент “identifier” та обов'язковий атрибут “gml:id”. Ці властивості успадковуються усіма компонентами, якими можна замінити AbstractTimeObject.

“TM\_Primitive” подають елементом об'єкту “AbstractTimePrimitive”.

“TM\_GeometricPrimitive” подають елементом об'єкту “AbstractTimeGeometricPrimitive”.

“TM\_TopologicalPrimitive” подають елементом об'єкту “AbstractTimeTopologyPrimitive”.

“TM\_Complex” подають елементом об'єкту “AbstractTimeComplex”.

“TM\_TopologicalComplex” подають елементом об'єкту “AbstractTimeTopologyComplex”.

## Г.2.5.3 Конкретні часові геометричні примітиви

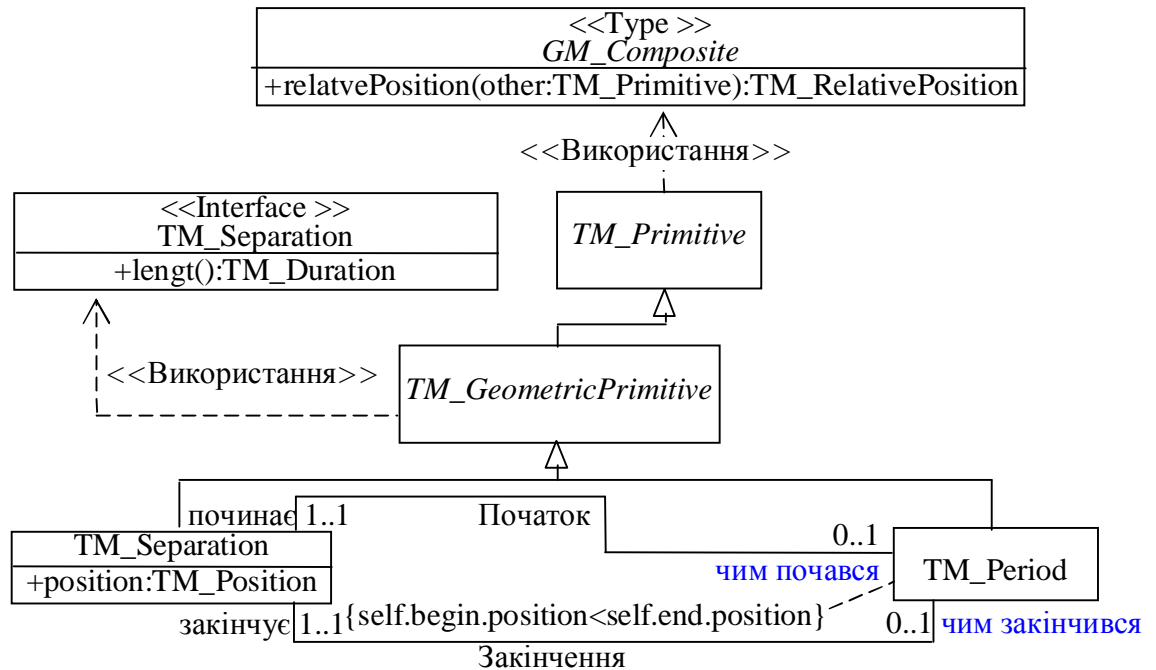


Рисунок Г.31 – Профіль часових геометричних об’єктів, адаптованих із стандарту ISO 19108

“`TM_Primitive`” подають елементом об’єкту “`AbstractTimePrimitive`”. Були додані додаткові властивості “`relatedTime`”, якими подають результат операцій “`relativePosition(other:TM_Primitive)`”.

“`TM_GeometricPrimitive`” подають елементом об’єкту “`AbstractTimeGeometricPrimitive`”. Була додана додаткова властивість “`abstractTimeLength`”, якою подають результат операції “`length()`”.

“`TM_Instant`” подають елементом об’єкту “`TimeInstant`”. Атрибут “`position`” подають властивістю “`timePosition`”.

“`TM_Period`” подають елементом об’єкту “`TimePeriod`”. Ролі “початок” (“`begin`”) та “кінець” (“`end`”) подають властивостями асоціації того ж імені у GML. Ці асоціації мають альтернативне подання в GML таким чином: “кінець” є у блоці вибору з “`endPosition`” та “початок” із “`beginPosition`”; останнє в кожному випадку має простий контент, як розглянуто в підпункті 14.2.2.5.

## Г.2.5.4 Часова тривалість

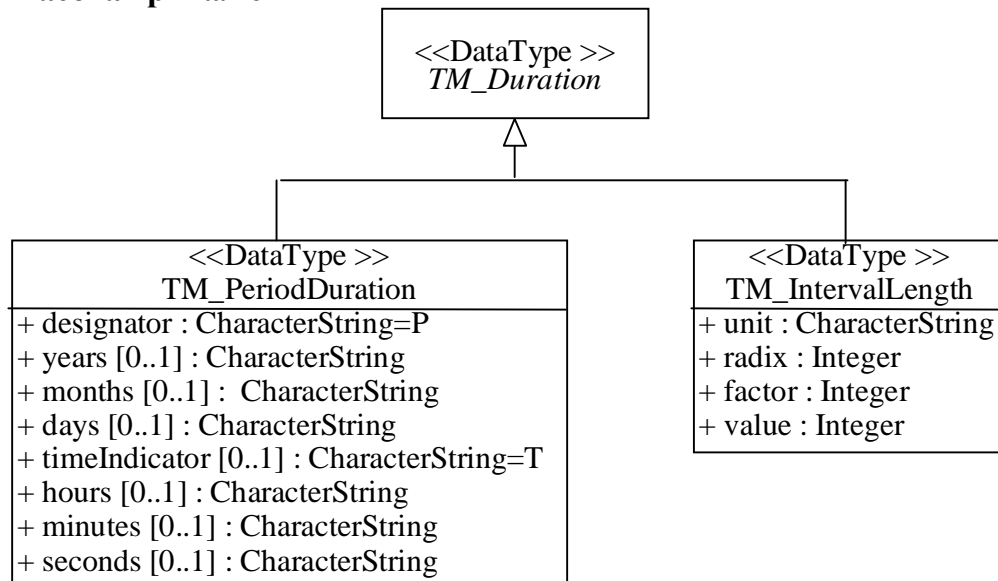


Рисунок Г.32 – Типи даних, якими подають часову тривалість у ISO 19108

GML елемент властивості “abstractTimeLength” є абстрактним, або з елементом “timeInterval”, або з елементом “duration”, що його замінюють. Вони мають XML типи схеми, якими реалізують типи иданих, показані на рисунку Г.32 таким чином:

“TM\_IntervalLength” реалізовано з використанням типу простого контенту, побудованого шляхом додавання XML атрибутів “unit”, “radix” та “factor” до вбудованого типу “decimal” XML схеми;

“TM\_PeriodDuration” реалізовано вбудованим типом “duration” XML схеми (див. розгляд у підпункті 14.2.2.8). Тип “duration” XML схеми передбачає літерне значення з лексичною формою, описане в ISO 8601, яке усуває необхідність окремо реалізовувати список атрибутів класу TM\_PeriodDuration.



## Г.2.5.5 Часова позиція

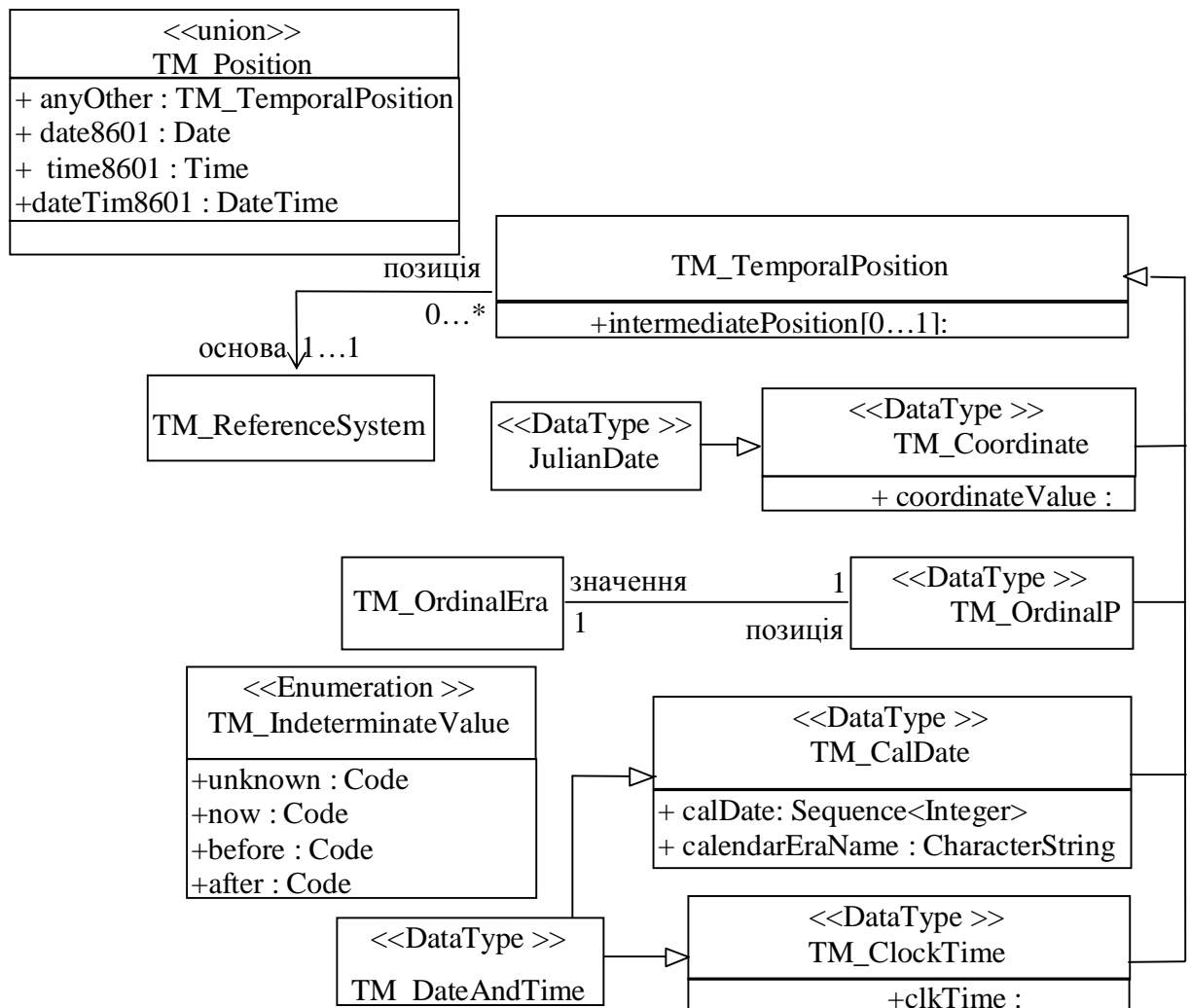


Рисунок Г.33 – Часова позиція із стандарту ISO 19108

Компоненти, подані як UML атрибути типу TM\_Position, подають як GML властивості з типом XML схеми gml:TimePositionType. Це є типом простого контенту, подробиці отримання якого описані в підпункті 14.2.2.5. Ним подають вимоги, показані на рисунку 33 таким чином:

- “TM\_Coordinate”, що надає часову позицію, подану одиничним числом, реалізовано типом “decimal” XML схеми;
- “TM\_OrdinalPosition”, що забезпечує асоціацію з TM\_OrdinalEra, реалізовано типом “anyURI” XML схеми, що відповідає шаблону, використаному в GML, де асоціації реалізовані шляхом посилань;
- “TM\_CalDate”, що має атрибути, які складаються з послідовності чисел для календарної дати, та імені ери, реалізовано у gml:CalDate об’єднанням (вибором) типів “date”, “gYear”, “gYearMonth” XML схеми, лексичні подання яких відповідають ISO 8601, до яких доданий XML атрибут “calendarEraName”;

- “TM\_ClockTime”, що включає послідовність чисел, які описують випадок, що повторюється щоденно, реалізовано типом “time” XML схеми, лексичне подання якого відповідає ISO 8601;
- “TM\_DateAndTime” реалізовано типом “dateTime” XML схеми, лексичне подання якого відповідає ISO 8601;
- варіанти “date8601”, “time8601” та “dateTime8601”, показані у TM\_Position, реалізовано уже реалізованими типами “date”, “time” та “dateTime” XML схеми;
- “IndeterminatePosition” подають із використанням XML атрибуту такого ж імені;
- роль “основа (“frame”)” реалізовано з використанням XML атрибуту такого ж імені, значення якого має тип “anyURI”, який відповідає шаблону, використаному в GML, де асоціації реалізовані шляхом посилань.

### Г.2.5.6 Часова топологія

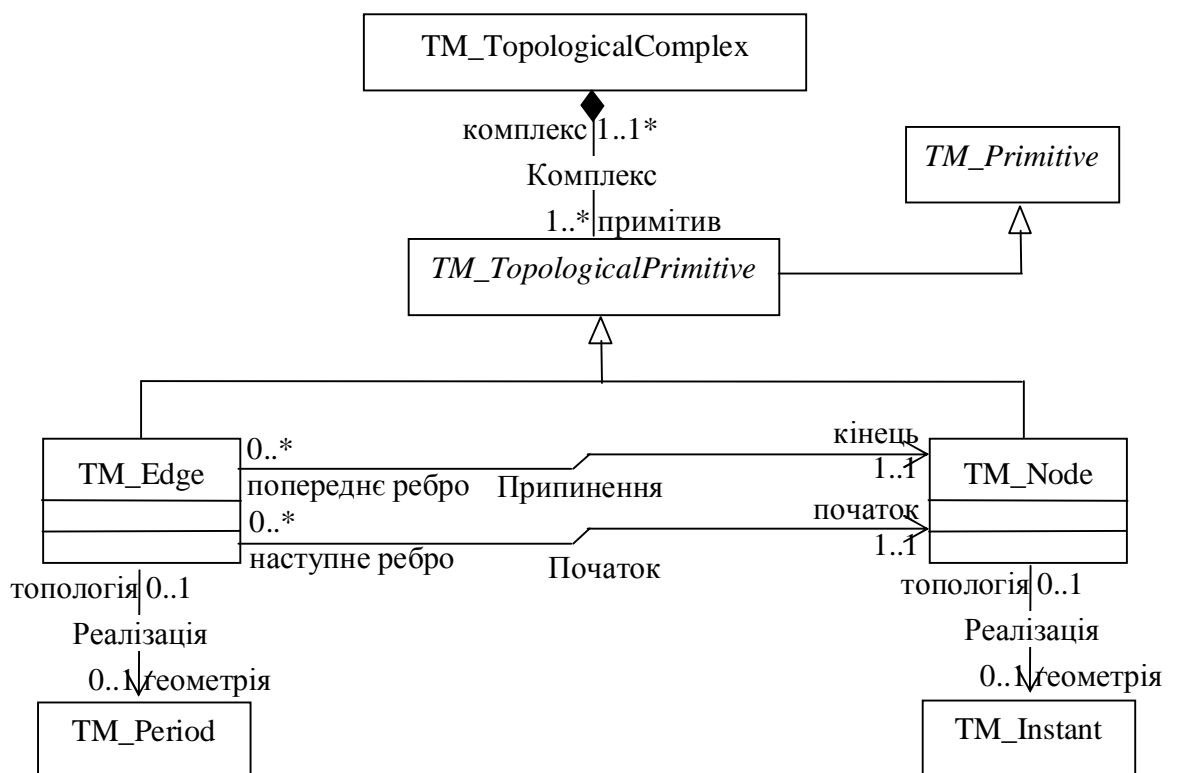


Рисунок Г.34 – Профіль часової топології, адаптований із стандарту ISO 19108

“TM\_TopologicalComplex” подано елементом об’єкту “AbstractTimeTopologyComplex”. Роль “примітив” (“primitive”) реалізовано елементом властивості того ж імені.

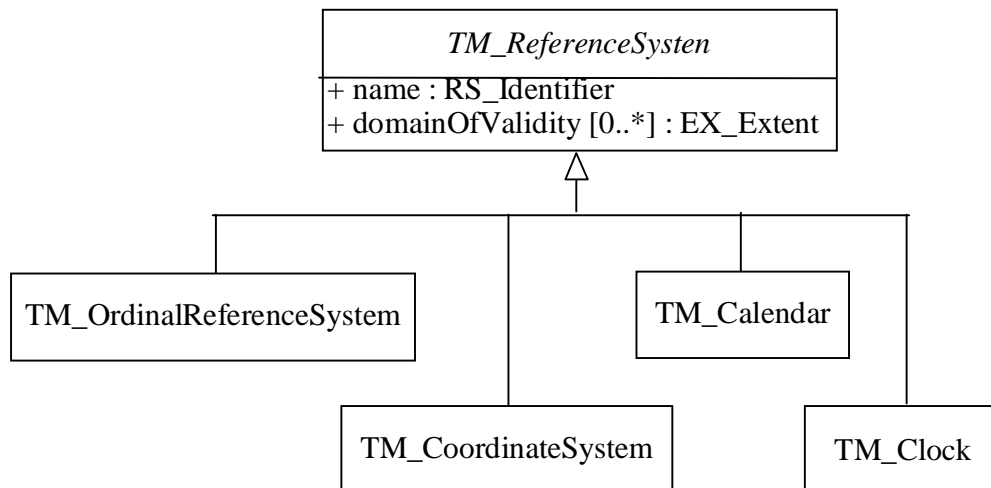
“TM\_TopologicalPrimitive” подано елементом об’єкту “AbstractTimeTopologyPrimitive”. Роль “комплекс” (“complex”) реалізовано як посилання на елемент властивості того ж імені, хоча це не є обов’язковим.

“TM\_Node” реалізовано елементом об’єкту “TimeNode”. Ролі “попереднє ребро” (“previousEdge”) та “наступнє ребро” (“nextEdge”) реалізовано елементами властивості того ж імені у GML. Роль “geometry” реалізовано властивістю “position”.

“TM\_Edge” реалізовано елементом об’єкту “TimeEdge”. Ролі “початок” (“start”) та “кінець” (“end”) реалізовано елементами властивості того ж імені у GML. Роль “геометрія” (“geometry”) реалізовано властивістю “extent”.

#### Г.2.5.7 Часові референцні системи

На діаграмах UML класів на рисунках Г.35-Г.38 показано профіль пакету “Temporal Reference Systems” (доречно порівняти з ISO 19108:2002, рисунки 7-10).



**Рисунок Г.35** – Ієрархія часових референцних систем із стандарту ISO 19108

“TM\_ReferenceSystem” реалізовано елементом об’єкту “AbstractTimeReferenceSystem”. Елемент “AbstractTimeReferenceSystem” може мати додаткові властивості: необов’язковий елемент “description”, один чи більше елементів “name”, необов’язковий елемент “identifier” та обов’язковий атрибут “gml:id”. Атрибут “domainOfValidity” реалізовано з використанням XML атрибуту того ж імені. Він має тип “рядок” XML схеми, який реалізує атрибут “description” для EX\_Extent (див. ISO 19115).

Ці властивості успадковані усіма компонентами, якими можна замінити AbstractTimeReferenceSystem.

## Г.2.5.8 Календарі та годинники

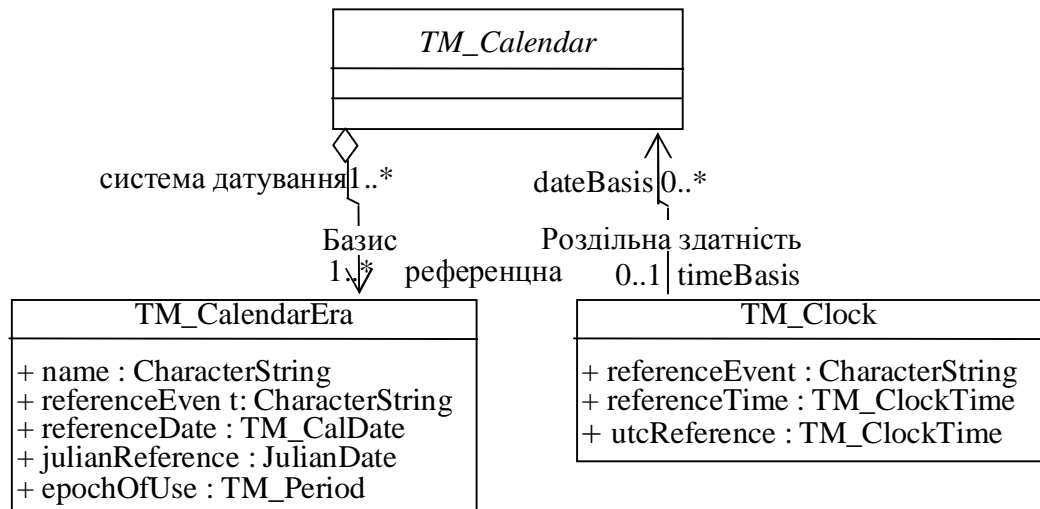


Рисунок Г.36 – Модель для календарів та годинників із стандарту ISO 19108

“TM\_Calendar” реалізовано елементом об’єкту “TimeCalendar”. Роль “referenceFrame” реалізовано як елемент властивості того ж імені.

“TM\_CalendarEra” реалізовано елементом об’єкту “TimeCalendarEra”. Атрибути “referenceEvent”, “referenceDate”, “julianReference” та “epochOfUse” реалізовано елементами властивості тих же імен.

“TM\_Clock” реалізовано елементом об’єкту “TimeClock”. Атрибути “referenceEvent”, “referenceTime” та “utcReference” реалізовано елементами властивості тих же імен. Роль “dateBasis” реалізовано елементом властивості того ж імені.

## Г.2.5.9 Часові системи координат

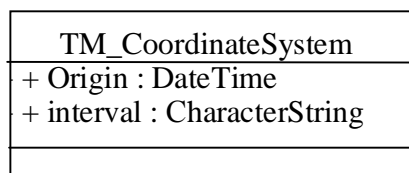
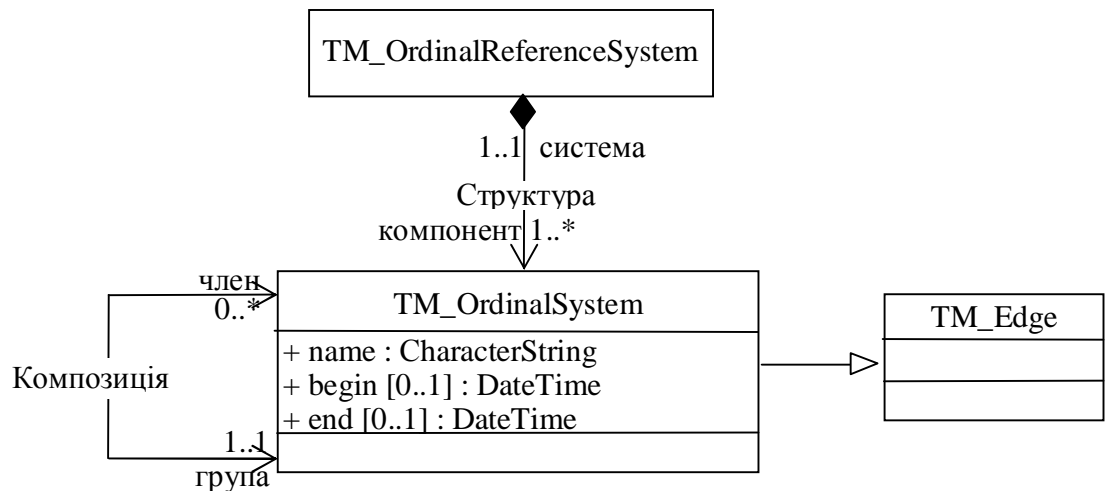


Рисунок Г.37 – Модель для часової системи координат, із стандарту ISO 19108

“TM\_CoordinateSystem” реалізовано об’єктом елементу “TimeCoordinateSystem”. Атрибут “origin” реалізовано як вибір елементів властивості “origin”, що посилається на TimeInstant, або “originPosition”, що безпосередньо кодує позицію. Атрибут “interval” реалізований як елемент властивості того ж імені, з використанням TimeIntervalLengthType, що відповідає стандарту ISO 11404.

### Г.2.5.10 Часова порядкова референцна система



**Рисунок Г.38** – Модель для часової порядкової референцної системи, адаптована із стандарту ISO 19108

“TM\_OrdinalReferenceSystem” реалізовано елементом об’єкту “TimeOrdinalReferenceSystem”. Роль “компонент” (“component”) реалізовано як елемент властивості того ж імені.

“TM\_OrdinalEra” реалізовано елементом об’єкту “TimeOrdinalEra”. Атрибути “name”, “begin” та “end” реалізовано властивостями “name”, “start” та “end”, успадкованими від TimeEdge. Роль “член” (“member”) реалізовано як елемент властивості того ж імені. Роль “група” (“group”) реалізовано як посилання елементом властивості того ж імені. Властивості – необов’язкова “description” та обов’язкова “gml:id” – також успадковано від TimeEdge.

#### Г.2.5.11 Відповідність

Правила, які забезпечують відповідність профілю до ISO 19108, описано в 19108:2002, розділ 2 та додаток А. Враховуючи критерії, визначені у розділі 2, GML як прикладна схема для передачі даних націлена на відповідність до А.1. Умови цього розділу відповідності забезпечені профілем, визначеним вище.

#### Г.2.6 Правила ISO 19109 для прикладної схеми

GML реалізує піднабір загальної моделі географічного об’єкту, визначений у ISO 19109.

Додатково GML реалізує розширення. Загальна модель географічного об’єкту стосується лише типів географічних об’єктів, тоді як прикладна схема (у GML чи UML) часто має справу з додатковими типами інформації. Прикладами є типи даних, типи переліків, типи об’єднань і т. і., що не визначені явним чином у загальній моделі географічного об’єкту. Тому за стандартом ISO 19109 можливим є тільки однобічний перехід із загальної моделі географічного об’єкту до прикладної схеми.

Подібно до випадку UML, перехід від загальної моделі географічного об’єкту до GML моделі географічного об’єкту взагалі є прямим.

Зміни, показані в таблиці Г.6, застосовані до загальної моделі географічного об’єкту ISO 19109.

Таблиця Г.6 – Опис реалізації ISO 19109

	Зміна	Пояснення
1	GF_FeatureOperation видалено	Операція не підтримується
2	Множинність для GF_InheritanceRelation/supertype змінена на 1	Тільки одиничне спадкування підтримується та може бути перекладене в механізм групи заміни XML схеми

Деякі додаткові коментарі по метакласах загальної моделі географічного об'єкту подані в таблиці Г.7.

Таблиця Г.7 – Зауваження з реалізації ISO 19109

	Зміна	Пояснення
1	GF_AssociationType	Взагалі, збирання ролей в асоціації в асоціацію не є безпосередньо поданим у GML прикладних схемах. Якщо відношення між двома ролями в асоціації треба подати явним чином у GML прикладній схемі, то ролі мають бути пов'язані перехресними посиланнями шляхом: - подання кваліфікованого імені цільового класу об'єктів в елементі анотації appInfo gml:targetElement, що належить до типу xsd:string; - подання кваліфікованого імені властивості зворотних ролей в асоціації в елементі анотації appInfo gml:reverseProperty, що належить до типу xsd:string; - необов'язкового подання імені асоціації в елементі анотації appInfo gml:associationName, що належить до типу which is of type xsd:string
2	GF_Constraint	Обмеження можна перекласти в обмеження Схема трону або просто подати як текст в анотаціях документації

Відношення між прикладними схемами в UML та в GML описано як частина додатків Д та Е, які стосуються переходу між ISO 19109 та GML прикладними схемами.

## Г.2.7 Просторова прив'язка по координатах відповідно до ISO 19111

### Г.2.7.1 Огляд

Повну концептуальну модель ISO 19111 реалізовано в GML. Перехід є у цілому прямим переходом від концептуальної моделі до GML схеми. Тільки компоненти схеми із складнішим переходом деталізовані в Г.2.7.

У всіх випадках, коли GML об'єкт не був асоційований з референчною системою координат, це реалізовано атрибутом (srsName), що вказує на елемент gml:AbstractCRS.

Додаткові атрибути визначені в моделі контенту того ж елементу, що забезпечує надлишкову інформацію про референцну систему координат. srsDimension є розмірністю референцної системи координат, згідно з визначенням референцної системи координат. Атрибути axisLabels та uomLabels є списками написів та одиниць вимірювання, асоційованих із різними осями референцної системи координат.

### Г.2.7.2 Пакет “ідентифікований об'єкт” (Identified Object)

Для реалізації в GML класи IO\_IdentifiedObjectBase та SC\_CRS об'єднано в клас IO\_IdentifiedObject, щоб підтримувати простіше XML кодування класу IO\_IdentifiedObject. Треба зауважити, що атрибут “name”, яким характеризують SC\_CRS, переходить в атрибут “name” класу IO\_IdentifiedObject.

Тип `IO_IdentifiedObject`, реалізований елементом `gml:IdentifiedObject` та його моделлю контенту, отримують шляхом розширення від `gml:DescriptionType`.

`RS_Identifier` реалізовано з використанням `gml:CodeType`. При використанні атрибут "version [0..1]" треба подавати в атрибуті `codeSpace`, яким характеризують `gml:CodeType`.

*Приклад 1.* Ім'я для референційної системи координат "4326" з версії 6.3 набору даних EPSG Міжнародної асоціації виробників нафти та газу можна подавати з використанням URN, визначеним в просторі імен `urn:ogc` таким чином:

```
<name codeSpace="urn:ogc:def:crs:EPSG:6.3:">4326</name>
```

Використання необов'язкової властивості `gml:description` підтримується для того, щоб дозволити кодування додаткової інформації про кожний об'єкт референційної системи координат.

Атрибут "name" реалізовано властивістю `gml:identifier`.

Атрибут "identifier" реалізовано властивістю `gml:name`.

Атрибут "alias" також реалізовано властивістю `gml:name`.

**Примітка 1.** При зворотному переході ідентифікатори були б тими `gml:name` разом із `codeSpace`, що можна перевести в `RS_Identifier`, та всі інші стали б альтернативними іменами.

### Г.2.7.3 Пакет референційної системи координат

Клас `SC_CRS` class реалізований з допомогою `gml:AbstractCRS`.

Роль "coordinateSystem"/"...CS" в асоціації з конкретним підкласом класу `CS_CoordinateSystem` реалізовано з допомогою `gml:coordinateSystem/gml:...CS`, де ім'я специфічного типу системи координат замінює еліпси.

Роль "дата" ("datum") в асоціації з конкретним підкласом класу `CD_Datum` реалізовано з допомогою `gml:...Datum`, де ім'я специфічного типу дати замінює еліпси.

**Примітка 1.** Зміна цього імені потрібна, тому що використовують глобальні імена, і інакше не можна визначити тип цільового класу у GML схемі.

Роль "складова система координат" ("componentReferenceSystem") в асоціації від класу `SC_CompoundCRS` до класу `SC_SingleCRS` реалізовано властивістю `gml:componentReferenceSystem`.

Асоціацію "базова референційна система координат" ("baseCRS") від класу `SC_ProjectedCRS` до класу `SC_GeodeticCRS` реалізовано властивістю `gml:baseGeodeticCRS`.

### Г.2.7.4 Пакет системи координат

Крім правил реалізації, визначених у пунктах Г.2.7.2 та Г.2.7.3, перехід є безпосереднім за виключенням того, що атрибут "axisUnitID", який характеризує `CS_CoordinateSystemAxis`, реалізований XML атрибутом  `uom`.

### Г.2.7.5 Пакет дат

Крім правил реалізації, визначених у пунктах Г.2.7.2 та Г.2.7.3, перехід є прямим за виключенням того, що порядок властивостей у `CD_Datum` змінено у переході до `gml:AbstractDatum`.

### Г.2.7.6 Пакет координатних операцій

Класи `CC_Conversion` та `CC_Transformation` реалізовано у два кроки як окремі абстрактні та конкретні елементи.

Ролі “значення параметру” (“parameterValue”) та “метод” (“method”) в асоціації від класу `CC_Operation` реалізовано в конкретних елементах `gml:Conversion` та `gml:Transformation` для зменшення потреби у використанні обмеження XML схеми.

Порядок властивостей у `CC_CoordinateOperation` змінено у переході до `gml:AbstractCoordinateOperation`.

Роль “координатна операція” (“coordOperation”) в асоціації від класу `CC_ConcatenatedOperation` до класу `CC_CoordinateOperation` реалізовано властивістю `gml:coordOperation`.

Ролі “параметр” (“parameter”) в асоціації до класів `CC_GeneralOperationParameter` або `CC_OperationParameter` реалізовано властивістю `gml:generalOperationParameter` або `gml:operationParameter`, залежно від цільового класу.

### Г.2.8 Просторова прив’язка по географічних ідентифікаторах ISO 19112

GML не надає наперед визначеної схеми для довідників. Однак, вона надає наперед визначені властивості для просторових прив’язок через географічні ідентифікатори:

- властивість `<gml:locationName>` містить текст, що описує позицію;
- властивість `<gml:locationReference>` посилається на текст, що описує позицію.

### Г.2.9 Метадані ISO 19115

GML не надає інформаційну модель для метаданих. Натомість, механізм включення метаданих або посилання на них надано для всіх елементів об’єктів.

**Примітка 1.** Як визначено у пункті 7.2.7, якщо метадані, що відповідають концептуальній моделі ISO 19115, треба закодувати в GML документі, для кодування інформації метаданих треба використовувати відповідну XML схему, визначену в ISO/TS 19139.

### Г.2.10 Кодування ISO 19118

Правила кодування, описані в Д.2, відповідають рівню 1 ISO 19118.

### Г.2.11 Покриття ISO 19123

UML модель GML профілю, визначена в цьому додатку, описує концептуальну модель абстрактних типів, визначених в ISO 19123. У таблиці Г.8 перекладено GML об’єкт покриття та імена властивостей в імена відповідних класів та їх атрибутів в ISO 19123 для полегшення порівняння з тим стандартом.

**Таблиця Г.8** – Опис реалізації ISO 19123

Засіб побудови ISO 19123	Засіб побудови GML
<code>CV_Coverage</code>	<code>AbstractCoverage</code> ( <code>AbstractCoverageType</code> )
<code>domainExtent</code> (атрибут)	<code>boundedBy</code> (властивість)
<code>domainElement</code> (ім'я ролі)	<code>domainSet</code> (властивість)
<code>rangeElement</code> (ім'я ролі)	<code>rangeSet</code> (властивість)
<code>атрибутValues</code>	<code>ValueArray</code> чи <code>AbstractScalarValueList</code>
<code>CV_CoverageFunction</code> (асоціація)	<code>coverageFunction</code> (властивість)
<code>CV_GridValuesMatrix</code>	<code>GridFunction</code>
<code>sequencingRule</code> (атрибут)	<code>sequenceRule</code> (властивість)
<code>CV_SequenceRule</code>	<code>SequenceRuleType</code>
<code>CV_SequenceType</code>	<code>SequenceRuleNames</code>
<code>scanDirection</code> (атрибут)	<code>order</code> (атрибут)
<code>startSequence</code> (атрибут)	<code>startPoint</code> (властивість)



Засіб побудови ISO 19123	Засіб побудови GML
CV_Grid	Grid
CV_GridEnvelope	GridEnvelope
low (атрибут)	low (властивість)
high (атрибут)	high (властивість)
CV_RectifiedGrid	RectifiedGrid
origin (атрибут)	origin (властивість)
offsetVectors (атрибут)	offsetVector чи їх набір (властивість)
CV_DiscretePointCoverage	MultiPointCoverage
CV_DiscreteCurveCoverage	MultiCurveCoverage
CV_DiscreteSurfaceCoverage	MultiSurfaceCoverage
CV_DiscreteGridPointCoverage	GridCoverage чи RectifiedGridCoverage

Додаткові зміни, показані в таблиці Г.9, застосовано до пакету покриттів ISO 19123.

**Таблиця Г.9** – Опис профілю ISO 19123

Зміна	Обґрунтування
Всі підкласи CV_ContinuousCoverage видалені	Нині не підтримується у GML
Асоціація Coordinate Reference System видалена deleted from CV_Coverage	Замінено на атрибути srsName (або frame) на об'єктах геометрії або часових об'єктах у домені. GML пакет покриттів дозволяє, щоб доменні об'єкти були в різних референцних системах координат (чи на референцних основах)
CV_CommonPointRule видалено	Нині не підтримується у GML
AttributeValues видалено	Замінено на вибір між GML аналогами: ValueArray або AbstractScalarValueList
CV_GeometryValuePair видалено	GML покриття кодує тільки доменно-діпазонний функціональний вигляд
CV_GridValuesMatrix видалено	Перехід від точок регулярної сітки до значень діапазону, включно з правилом послідовності і т. і. міститься в об'єкті (GridFunction), що не є успадкованим від Grid
CV_GridCell	Нині не підтримується у GML

На діаграмі UML класів на рисунку Г.39 показано профіль кореня покриття “Coverage root” (доречно порівняти з ISO 19123:2005, рисунок 2).

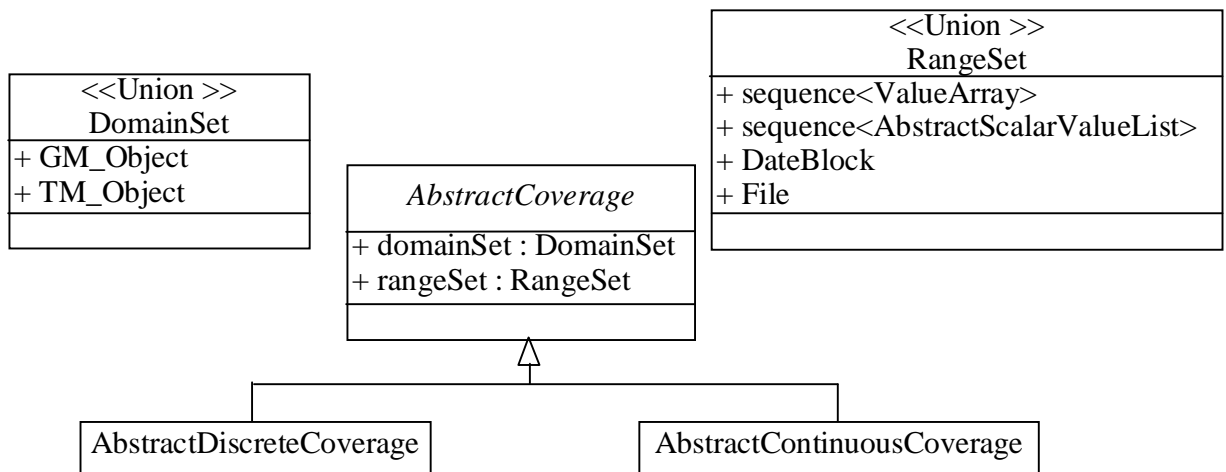


Рисунок Г.39 – Огляд покриттів

На діаграмі UML класів на рисунку Г.40 показано дискретні покриття (доречно порівняти з ISO 19123:2005, рисунки 3, 4 та 5).

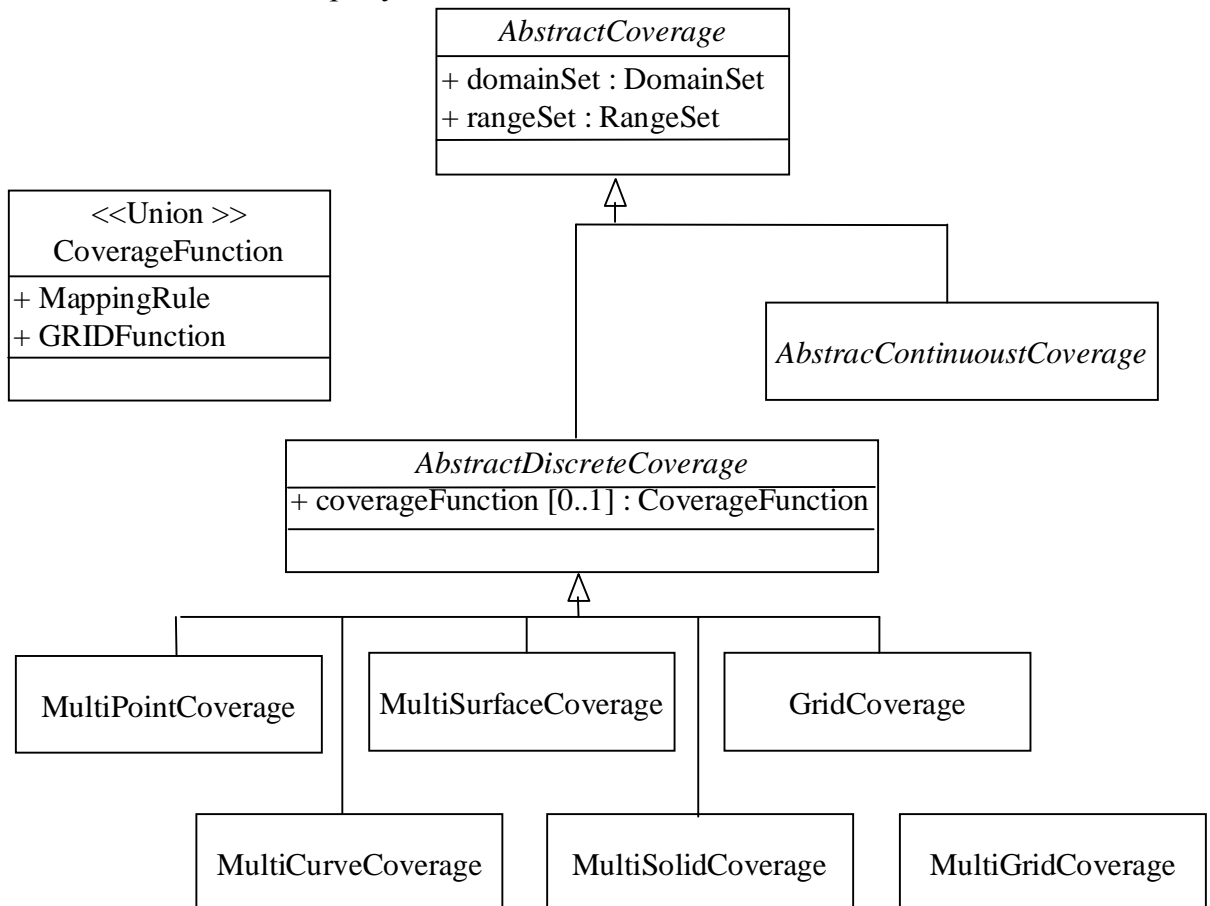


Рисунок Г.40 – Дискретні покриття

Правила, що визначають відповідність профілю ISO 19123, описані в ISO 19123:2005, розділ 2 та додаток А.

Дотримані умови таких розділів відповідності:

- А.1.1 Інтерфейс простого покриття;
- А.1.2 Інтерфейс дискретного покриття.
- А.1.4 Інтерфейс квадрилатерального GRID-покриття.

Треба зауважити, що похідні атрибути розглядають як операції та припускають, що похідні атрибути будуть отримані з даних аплікацією, яка працює з GML екземплярами.

### **Г.3 Розширення профілю комплексу міжнародних стандартів 19100**

#### **Г.3.1 Огляд**

У наступних підрозділах визначені додаткові частини GML, не охоплені профілем, визначеним у Г.2. UML використовують як мову концептуальних схем для опису додаткових елементів у відповідності до ISO/TS 19103. Подробиці семантики додаткових класів викладено в розділах 7-19.

GML компоненти схеми згруповано, виходячи з семантики, на основі структури розділів 7-19, і були створені пакети для кожного угруповання. Потрібні додаткові класи, не задокументовані в Г.2, визначено в відповідному пакеті. Ці пакети є частиною пакету “GML”.

#### **Г.3.2 Пакет “basicTypes”**

Крім цих типів із стандарту ISO/TS 19103, у підрозділі 8.2 визначена низка додаткових типів, які потрібні та використовуються в інших документах GML схем.

Більшість із цих доповнень є результатом можливості надання інформації про порожню інформацію (атрибути nilReason). Її було додано до GML схеми на підставі вимог користувача та оскільки концепція була призначена для загального використання (див. рисунки 41, 42).

Типи списку – це лише типи зручності для спрощення написання GML схеми.

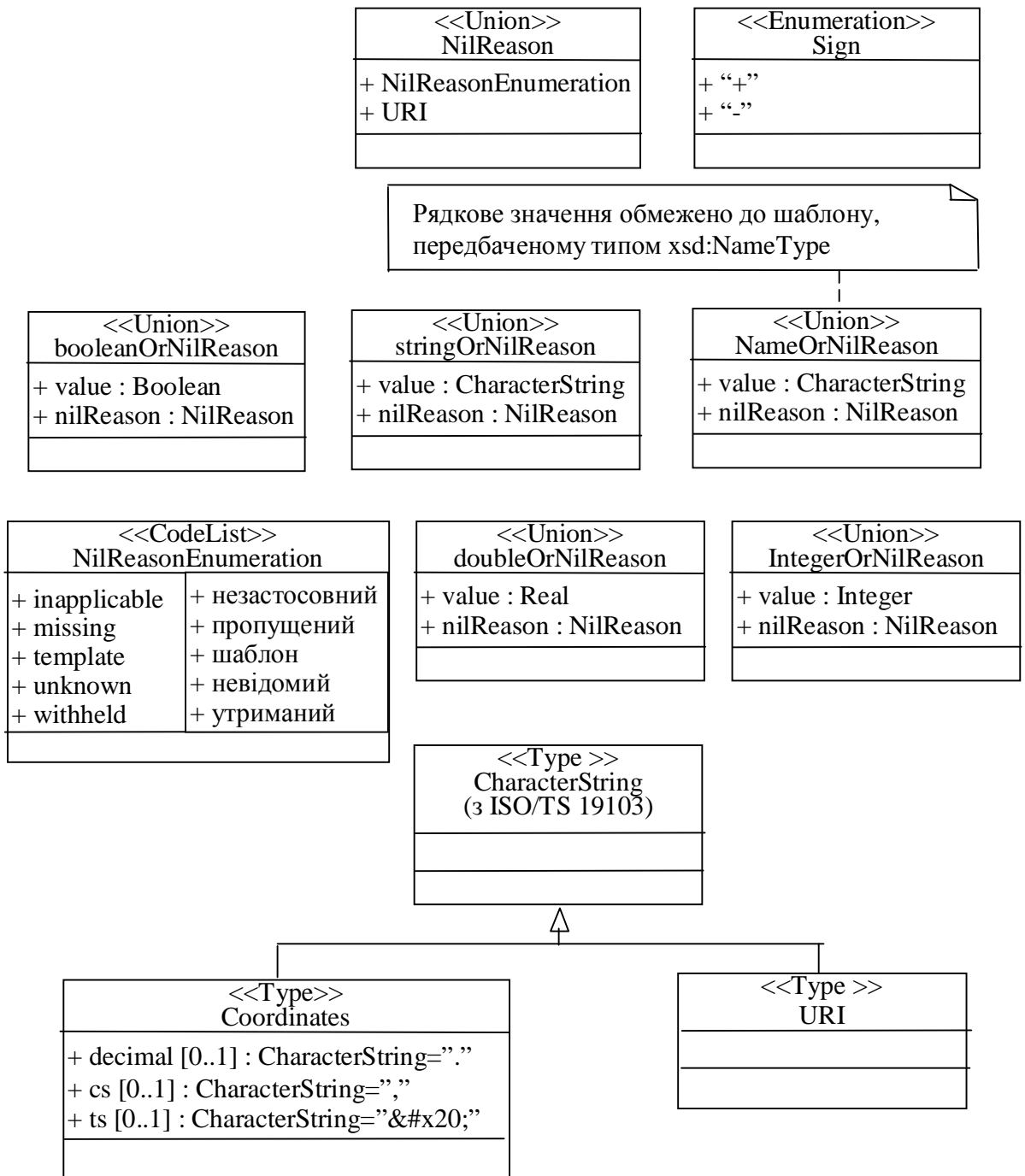


Рисунок Г.41 – Прості типи

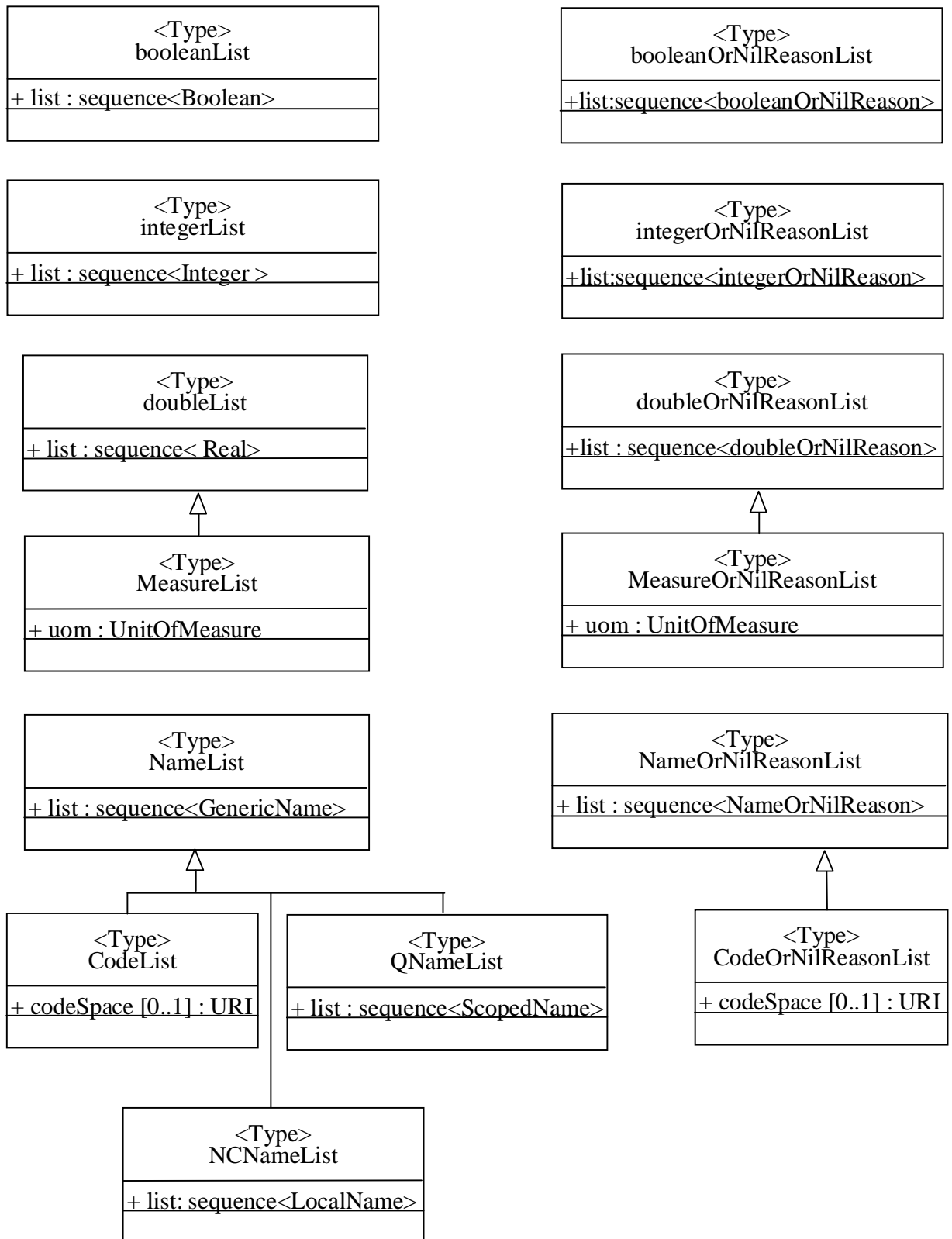


Рисунок Г.42 – Списки

### Г.3.3 Пакет “gmlBase”

У доповнення до цих базових типів із стандарту ISO/TS 19103, у підрозділі 7.2 визначено кілька додаткових типів, використаних іншими документами GML схем, крім тих, що є частиною переходу від концептуальної схеми до реалізації XML схеми. Подання типів об'єктів, ролей в асоціації і т. і. описане в додатку Д.

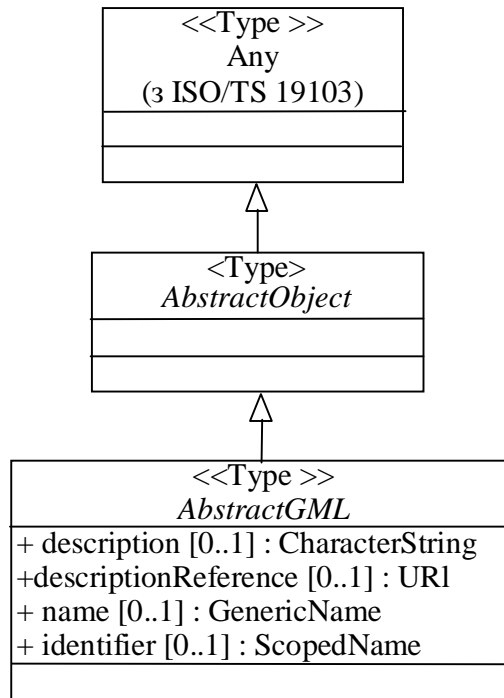


Рисунок Г.43 – Базові типи

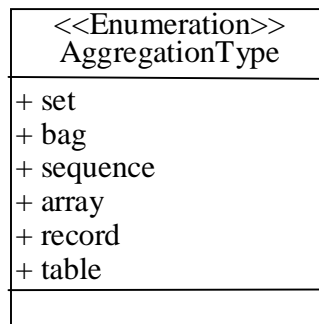


Рисунок Г.44 – Агрегатні типи колекцій

### Г.3.4 Пакет “географічний об’єкт” (“feature”)

У цьому підрозділі визначені додаткові типи, використані в розділі 9.

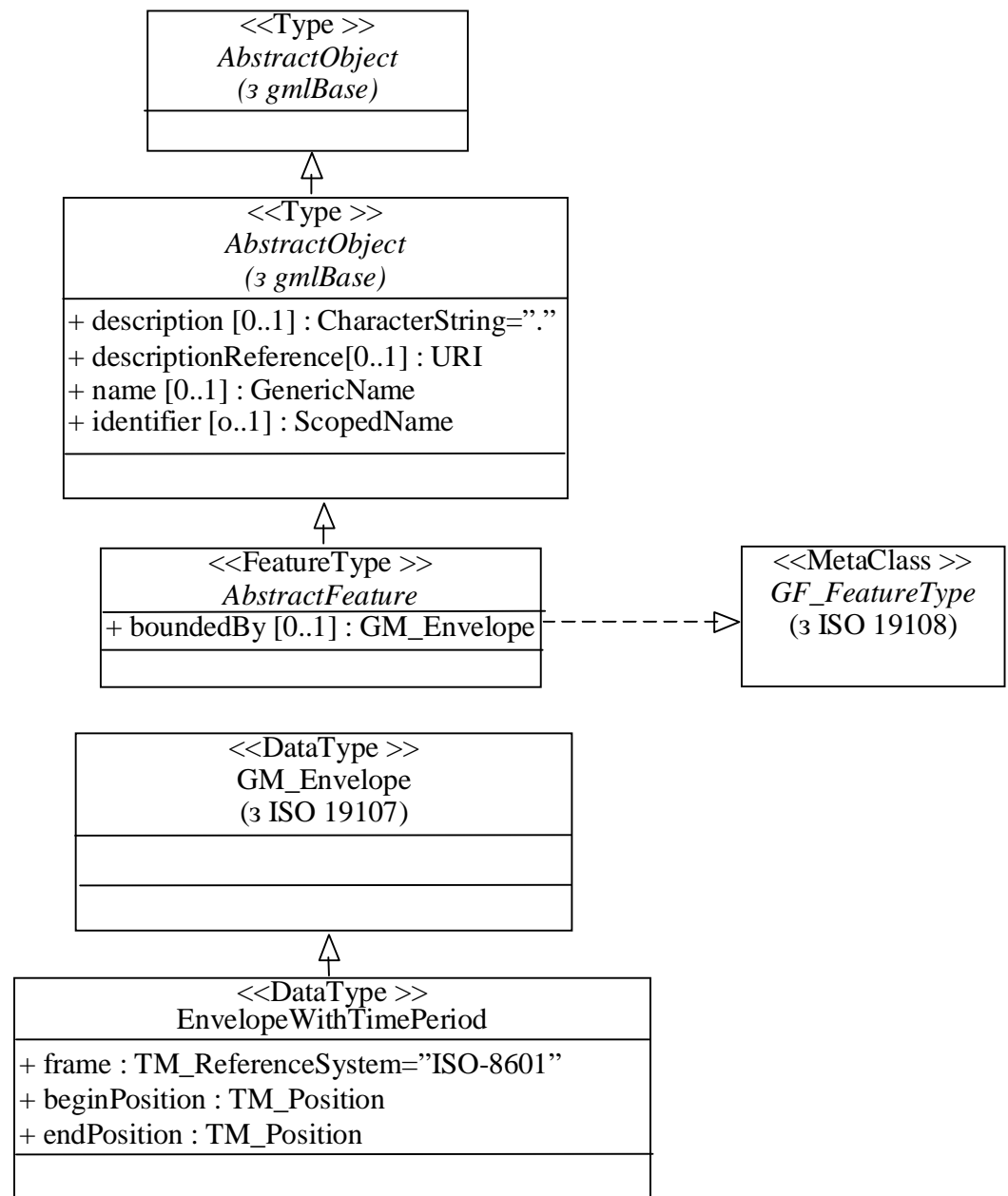


Рисунок Г.45 – Географічні об'єкти

Типи допомагають пояснити подання типів географічних об'єктів у GML. Крім тільки наперед визначених елементів, властивості визначені в цьому розділі. Всі інші концепції, розглянуті в розділі, є частиною переходу від концептуальної схеми до впровадження XML схеми (докладніше див. додаток Д).

### Г.3.5 Пакет “geometryBasic0d01”

У цьому підрозділі визначені додаткові типи, використані в пунктах 10.2.2 та 10.2.3.

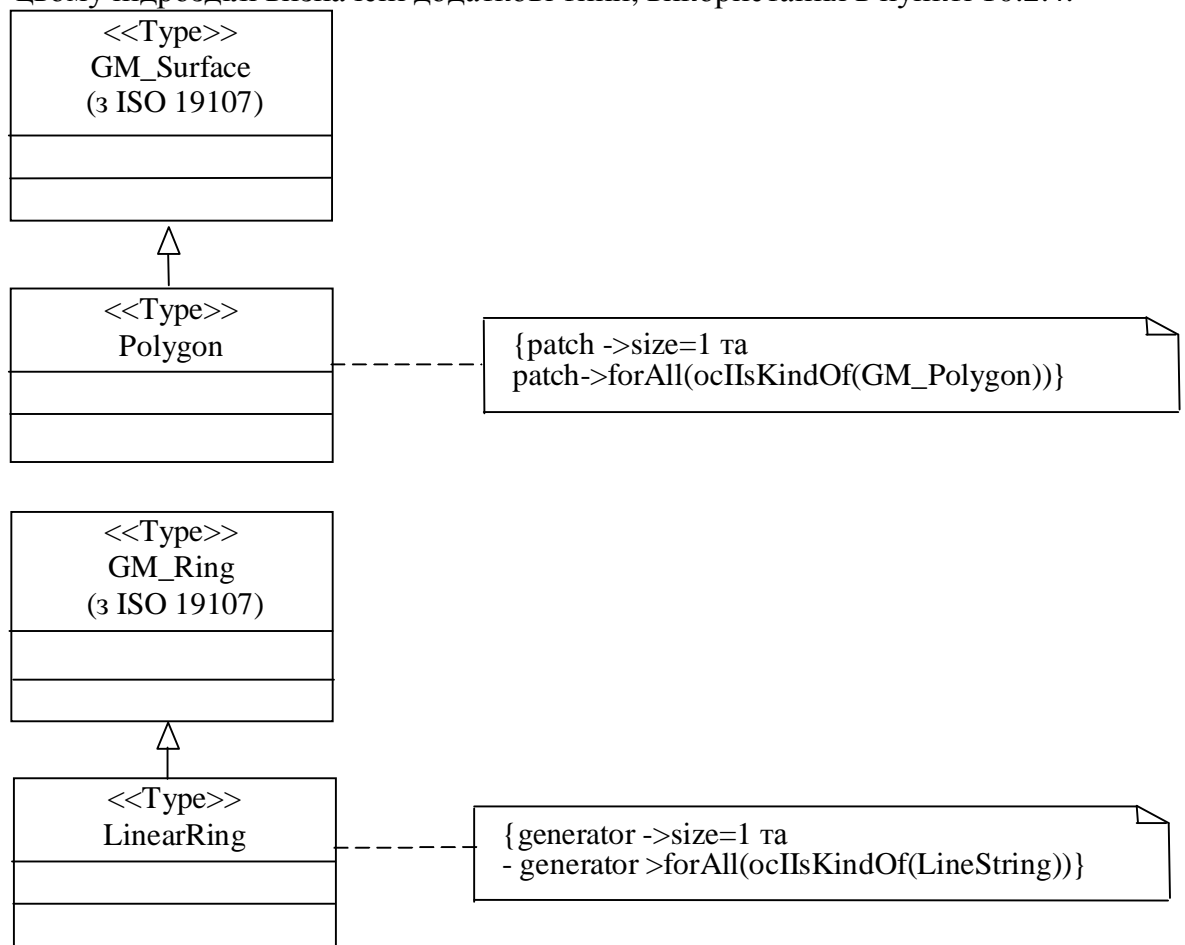


**Рисунок Г.46** – GM\_Curve із одним сегментом GM\_LineString

Додатковий підтип для GM\_Curve, доданий у GML. “LineString”, є спеціальною кривою, що складається тільки з відрізків GM\_LineString. XML подання елементу об’єкту “LineString” об’єднує всі відрізки в один відрізок, а його контрольні точки подані як безпосередні властивості, якими характеризують “LineString”.

### Г.3.6 Пакет “geometryBasic2d”

У цьому підрозділі визначені додаткові типи, використання в пункті 10.2.4.



**Рисунок Г.47** – GM\_Polygon з однією ділянкою GM\_Polygon



Додатковий підтип для GM\_Surface додано у GML. “Polygon” є спеціальною поверхнею, що складається тільки з однієї ділянки GM\_Polygon. Оскільки тільки існує одна ділянка, XML подання елемента об’єкту “Polygon” не враховує рівень ділянки, а властивості зовнішніх та внутрішніх меж ділянки подані як безпосередні властивості, якими характеризують “Polygon”.

Подібним чином “LinearRing” є одним кільцем (GM\_Ring подають одним лінійним рядком).

### Г.3.7 Пакет “geometryPrimitives”

У цьому підрозділі визначені додаткові типи, використані в розділі 11.

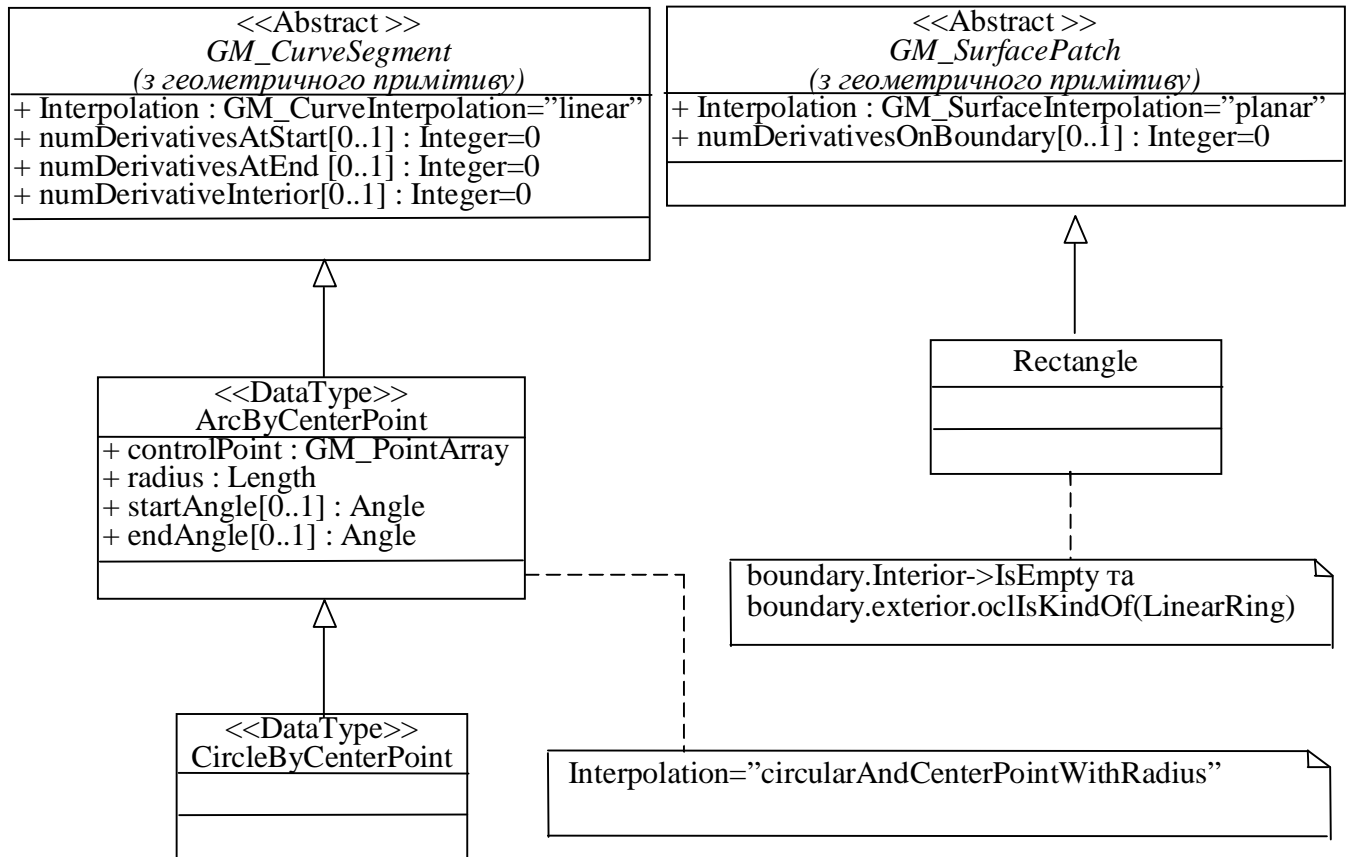


Рисунок Г.48 – Додаткові відрізки кривої та ділянки поверхні

Сегменти кривої “ArcByCenterPoint” та “CircleByCenterPoint”, як і ділянка поверхні “Rectangle”, були визначені в GML.

### Г.3.8 Пакет “geometryAggregates”

У цьому підрозділі визначені додаткові типи, використані в підрозділі 12.3.

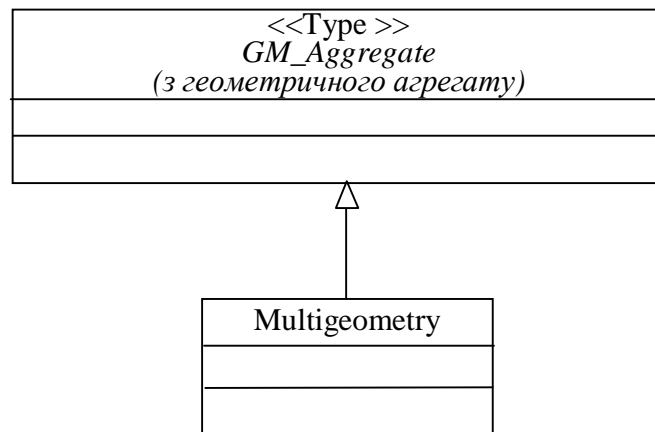


Рисунок Г.49 – Додаткові геометричні агрегати

У GML визначений геометричний агрегат, що має екземпляри, не обмежені до елементів тільки одного виміру: “MultiGeometry”.

### Г.3.9 Пакети “coordinateOperations”, “coordinateReferenceSystems”, “coordinateSystems”, “dataQuality”, “datums”, “referenceSystems”

#### Г.3.9.1 Огляд

У Г.3.10 визначено часові референсні системи як додатковий підтип референсних систем координат паралельно до типів для часових дат та часових систем координат.

Перехід до GML об’єктів, якими впроваджують ці типи (gml:TemporalCRS, gml:TimeCS та gml:TemporalDatum) є прямим та відповідає правилам, описаним у Г.2.7.

#### Г.3.9.2 UML схема пакету “coordinateReferenceSystems”

На рисунку Г.50 показано UML діаграму класу пакету референсні системи координат “coordinateReferenceSystems”, доречного для часової референсної системи координат.

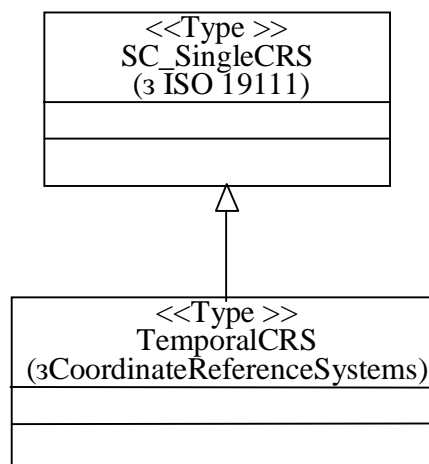


Рисунок Г.50 – Часові референсні системи

Таблиця Г.10 – Визначення елементів класу TemporalCRS

Опис:	Одновимірна референцна система координат, використана для запису часу
Стереотип:	(немає)
Походить від:	SC_SingleCRS
Ролі в асоціаціях:	Дата для TemporalDatum [1] Система координат coordinateSystem для CS_TimeCS [1] (асоціації, успадковані від SC_SingleCRS)
Публічні атрибути:	6 атрибутів, успадкованих від IO_IdentifiedObjectBase, RS_ReferenceSystem та SC_CRS

### Г.3.9.3 UML схема пакету "coordinateSystems"

Часова система координат є одновимірною системою координат, яка містить одну часову вісь та яку використовують для опису часової позиції точки у визначених одиницях часу відносно визначеного початку часу.

На рисунку Г.51 показана UML діаграма класу пакету системи координат "coordinateSystems", доречного для часових систем координат. Обмеження на асоціації між SC\_SingleCRS та CS\_CoordinateSystem показано на UML діаграмі класу на рисунку Г.52.

На UML діаграмі класу на рисунку Г.52 показано обмеження на асоціації між підтипами референцних систем координат.

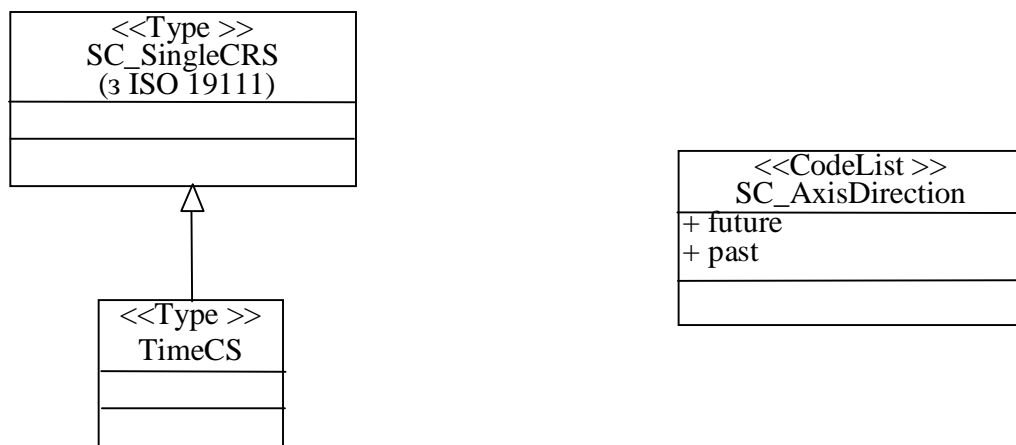
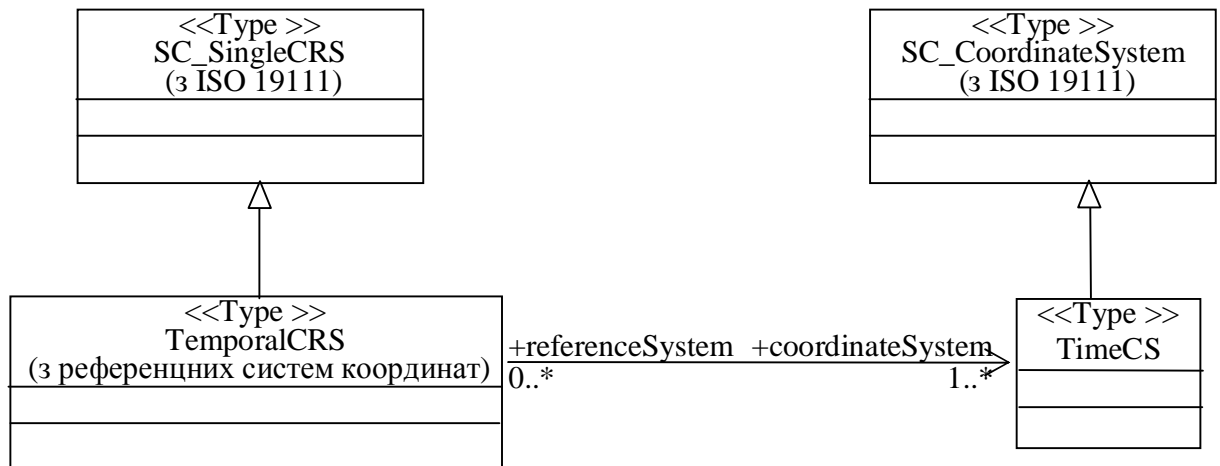


Рисунок Г.51 – Часові системи координат



**Рисунок Г.52** – Асоціація між часовою референчною системою координат та системою координат

**Таблиця Г.11** – Визначення елементів класу TimeCS

Description:	Одновимірна референчна система координат, що містить часову вісь, використану для опису часової позиції точки у визначених одиницях часу відносно визначеного початку часу. TimeCS має мати одну асоціацію осі.
Стереотип:	(немає)
Походить від:	CS_CoordinateSystem
Ролі в асоціаціях:	coordinateSystem для TemporalCRS [1]  (асоціації, успадковані від CS_CoordinateSystem)
Публічні атрибути:	4 атрибути, успадковані від IO_IdentifiedObject та IO_IdentifiedObjectBase

**Таблиця Г.12** – Визначення елементів класу CS\_AxisDirection

Description:	Напрямок додатного прирощення значень координат для вісі системи координат				
Стереотип:	(відсутнє)				
Походить від:	(відсутнє)				
Ролі в асоціаціях:	(відсутні)				
Чим використовується	CS_CoordinateSystemAxis				
Публічні атрибути:	CS_CoordinateSystemAxis				
Ім'я атрибуту	UML ідентифікатор	Тип даних	Обов'язковість	Множинність	Опис атрибуту
future	Future	CharacterString	Умовний	1	Додатний напрямок вісі до майбутнього
past	Past	CharacterString	Умовний	1	Додатний напрямок вісі до минулого

Description:	Напрямок додатного прирощення значень координат для вісі системи координат
Умова:	треба надавати один і тільки один з перелічених атрибутів

### Г.3.9.4 UML схема пакету дат (“datums”)

На рис. Г.53 показано UML діаграму класів пакету дат, доречно для часових дат. Обмеження на асоціацію між SC\_SingleCRS та CD\_Datum показано на UML діаграмі класів на рисунку Г.54.

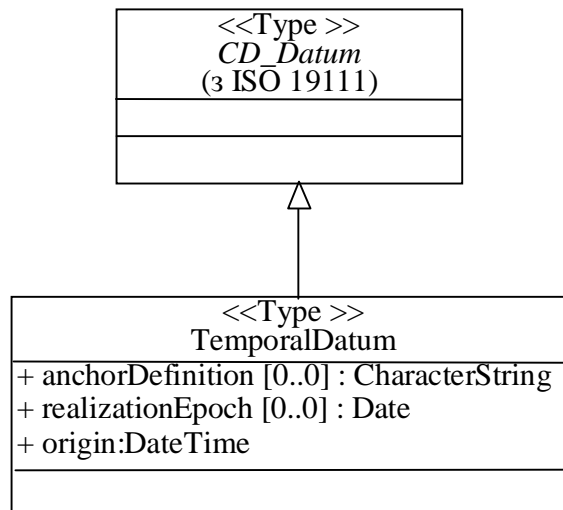


Рисунок Г.53 – Часова дата

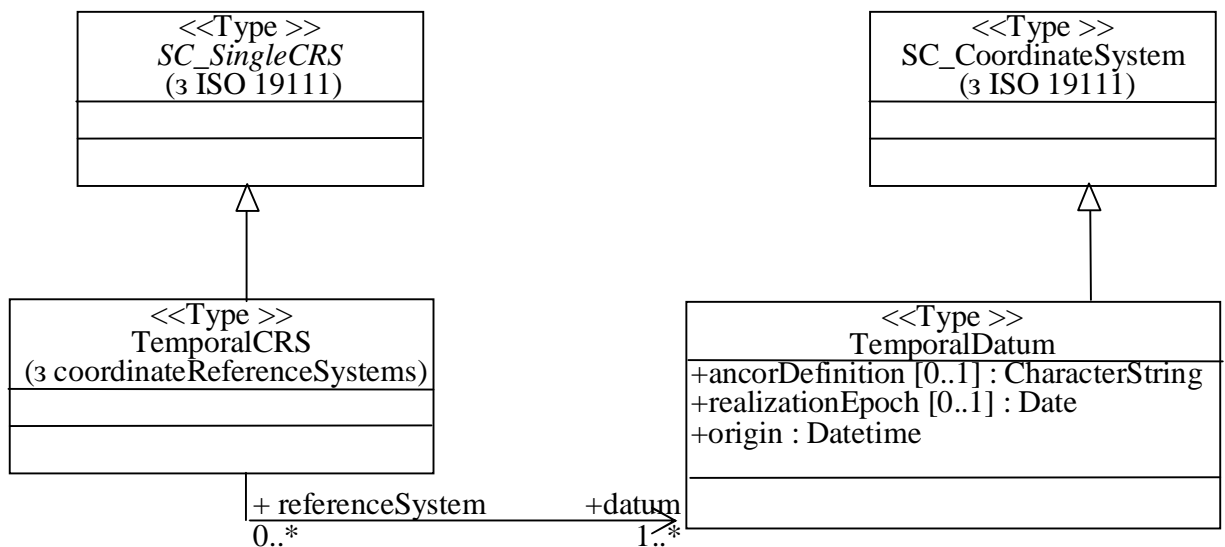


Рисунок Г.54 – Асоціація між часовою референсною системою координат та часовою датою

**Таблиця Г.13** – Визначення елементів класу TemporalDatum

Description:		Часова дата визначає початок часової референцної системи			
Стереотип:		(відсутнє)			
Походить від:		CD_Datum			
Ролі в асоціаціях:		Дата (datum) від TemporalCRS [1]			
Публічні атрибути:		8 атрибутів, успадкованих від CD_Datum, IO_IdentifiedObject та IO_IdentifiedObjectBase, плюс:			
Ім'я атрибуту	UML ідентифікатор	Тип даних	Обов'язковість	Множинність	Опис атрибуту
Початок	origin	DateTime	Обов'язковий	1	Дата та початок часу для часової дати
Із цих 8 успадкованих атрибутів змінені:					
Якірне визначення	anchorDefinition	Символьний рядок (CharacterString)	Обов'язковий	0	Часова дата не використовує цей атрибут
Епоха реалізації	realizationEpoch	Дата (Date)	Обов'язковий	0	Часова дата не використовує цей атрибут

**Г.3.10 Пакет “topology”**

У цьому підрозділі визначені додаткові правила, використані в розділі 13.

GML визначає кілька типів даних, що містять спрямовані топологічні примітиви (або посилаються на них), один на вимір: “TopoPoint”, “TopoCurve”, “TopoSurface” and “TopoVolume”. Вони успадковані та призначені для використання у властивостях географічних об'єктів.

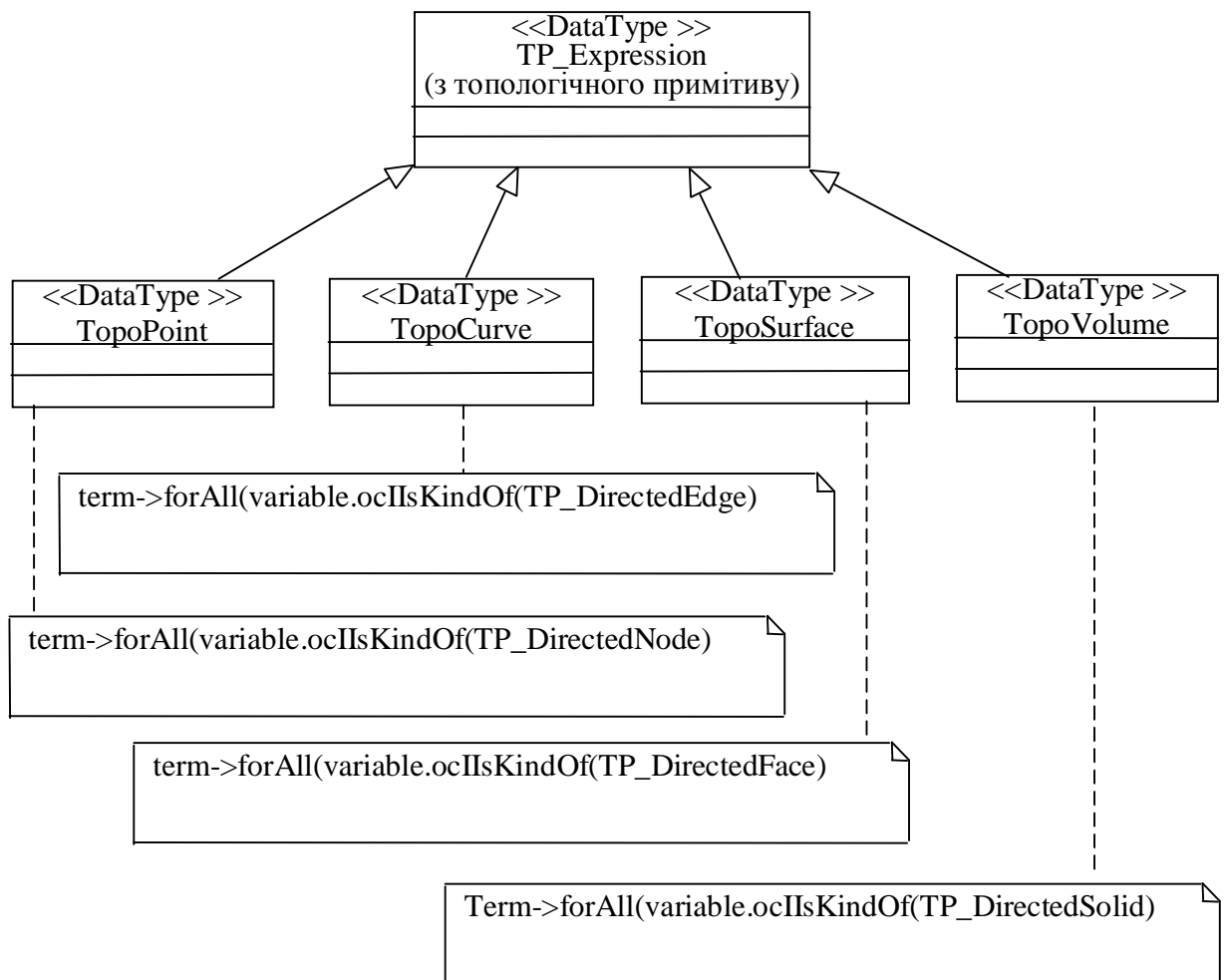


Рисунок Г.55 – Топологічні вирази

**Г.3.11 Пакет “dynamicFeature”**

У цьому підрозділі визначені додаткові типи, використані в підрозділі 14.5.

До GML додано концепцію динамічних географічних об'єктів, оскільки можливість подавати властивості, змінні в часі, розглядають як фундаментальну концепцію географічної інформації.

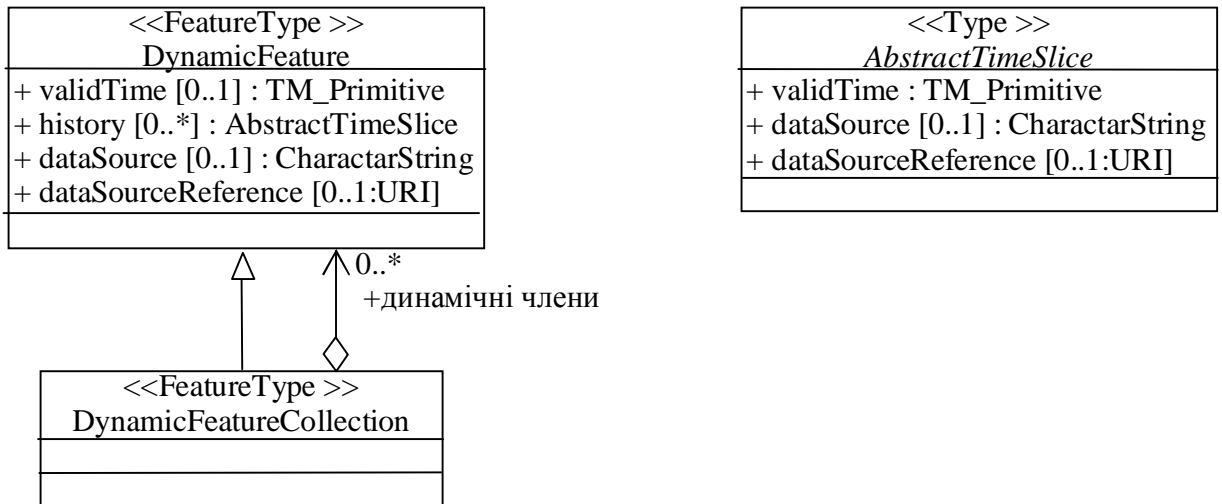


Рисунок Г.56 – Динамічні географічні об’єкти

### Г.3.12 Пакет “dictionary” (словник)

У цьому підрозділі визначені додаткові типи, використані в розділі 15. Концепцію словника було додано до GML, оскільки можливість кодувати словники списків кодів, одиниць та референцних систем координат є фундаментальною для роботи з даними екземплярів у прикладних схемах.

**Примітка 1.** Клас “Dictionary” треба називати GMLDictionary для уникнення конфлікту імен із класом із таким же іменем в ISO/TS 19103.

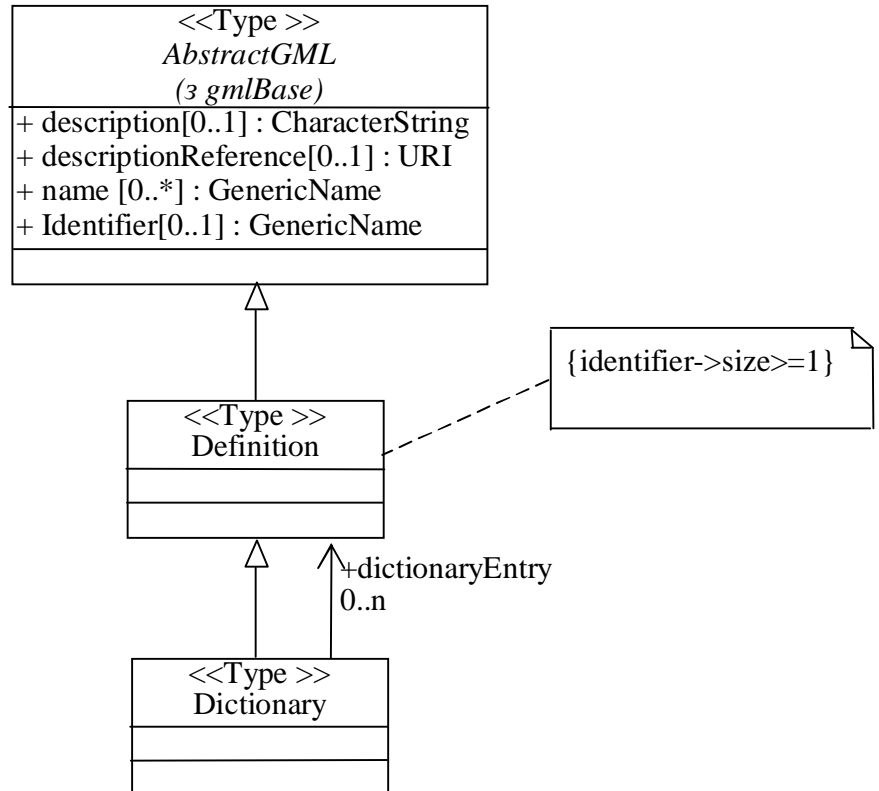
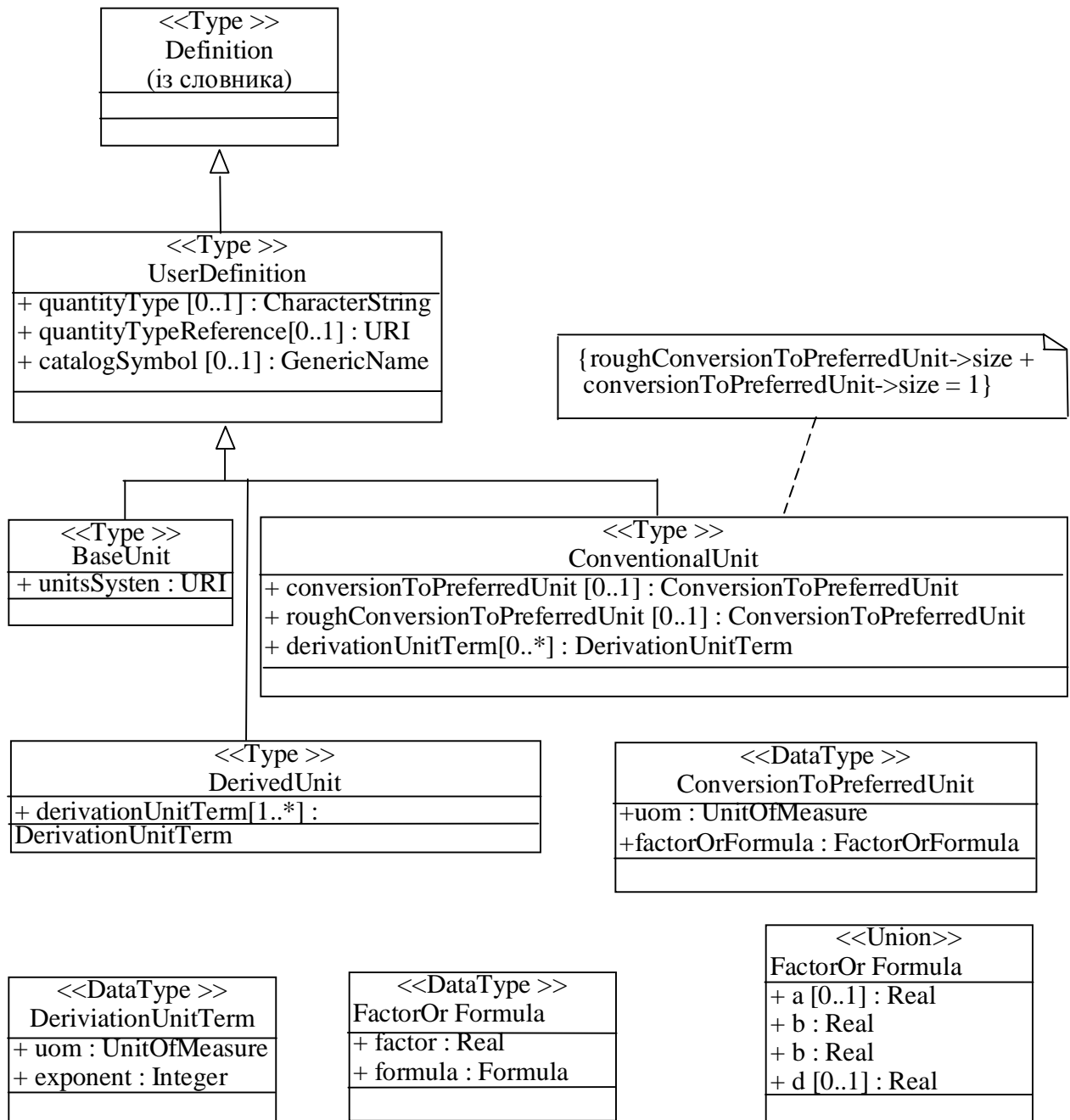


Рисунок Г.57 – Словники



**Г.3.13 Пакет “units” (одиниці)**

У цьому підрозділі визначені додаткові типи, використані в підрозділі 16.2.



**Рисунок Г.58** – Визначення одиниць

**Г.3.14 Пакет “measures” (міри)**

У цьому підрозділі визначені додаткові типи, використані в підрозділі 16.3. У додаток до підтипів “Міри” (“Measure”), як визначено в ISO/TS 19103, використовують інший підтип у контексті регулярних мереж, і тому його визначають як частину GML схеми (дивію рисунок Г. 59).

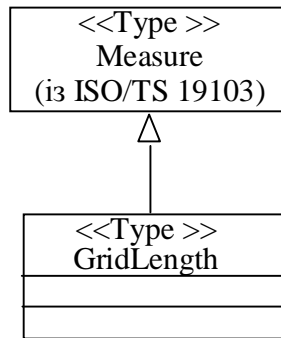


Рисунок Г.59 – Додаткові міри

**Примітка 1.** Для уникнення конфлікту імен класу Boolean в ISO/TS 19103, клас об’єктів булевських значень був названий BooleanValue.

### Г.3.15 Пакет “valueObjects”

У цьому підрозділі визначені додаткові типи, використані в підрозділі 16.4. Ці типи використовують у схемі спостережень (див. додаток Г.3.17).

Ієрархія компонентів показана в наступній UML діаграмі класів. UML відношення узагальнення використані для зазначення групи заміни XML схеми та членства в групі вибору. Відношення UML композиції використовують, щоб указати членство в моделі контенту типу XML схеми.

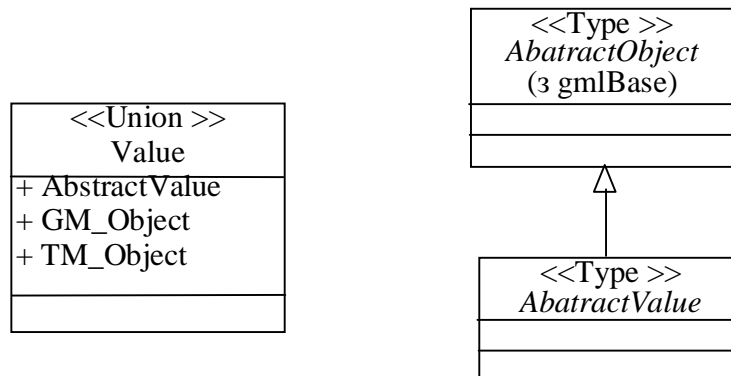


Рисунок Г.60 – Вирази значень

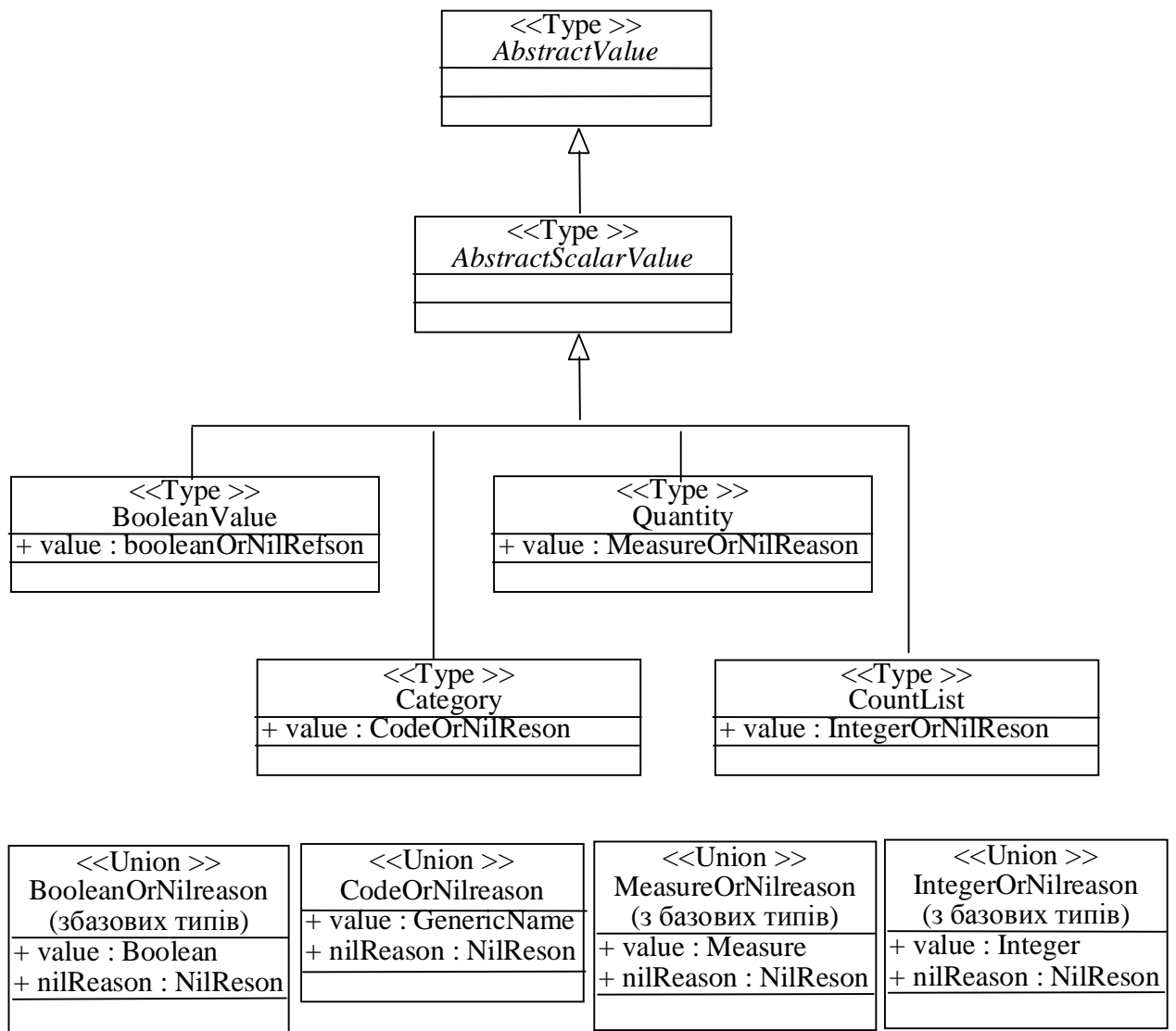


Рисунок Г.61 – Скалярні значення

**Примітка 1.** Для уникнення конфлікту з класом Boolean в ISO/TS 19103, клас об'єктів булевських значень названо BooleanValue.

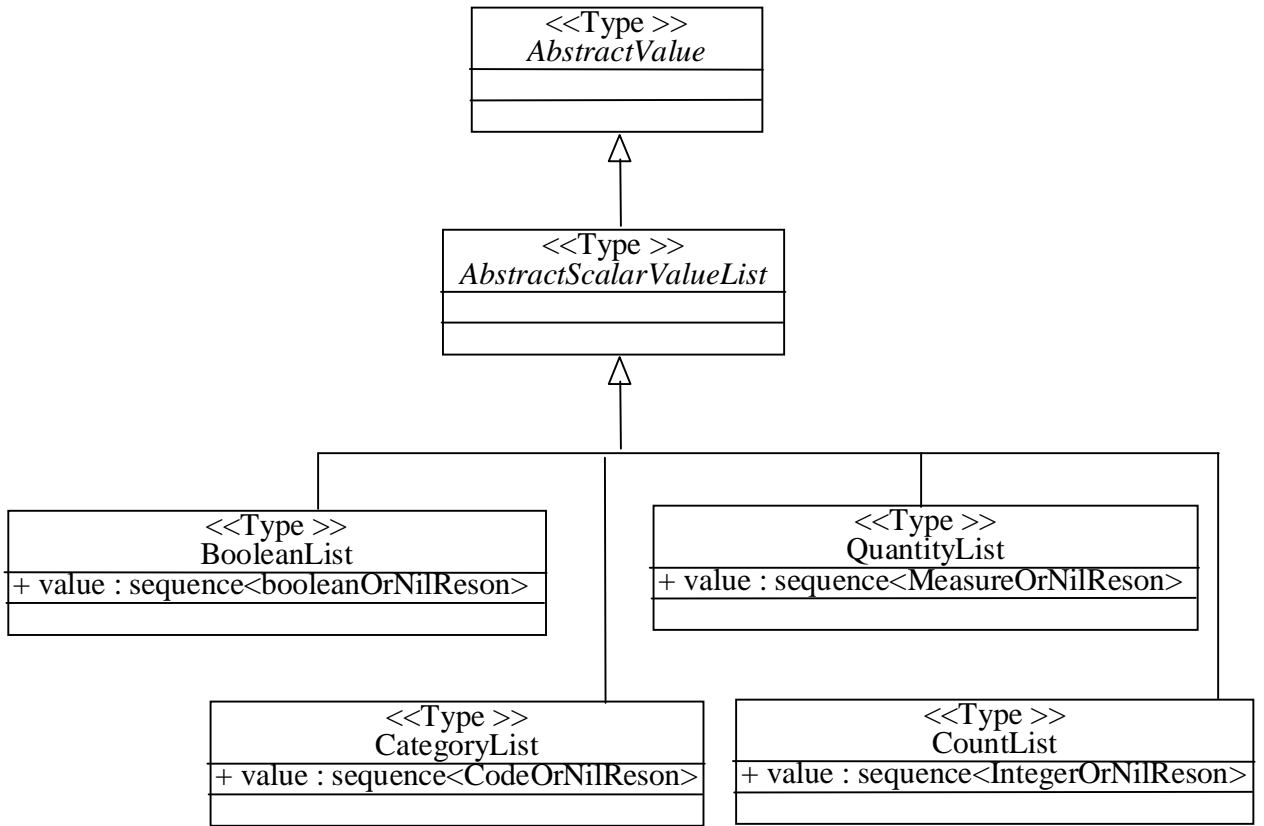


Рисунок Г.62 – Списки скалярних значень

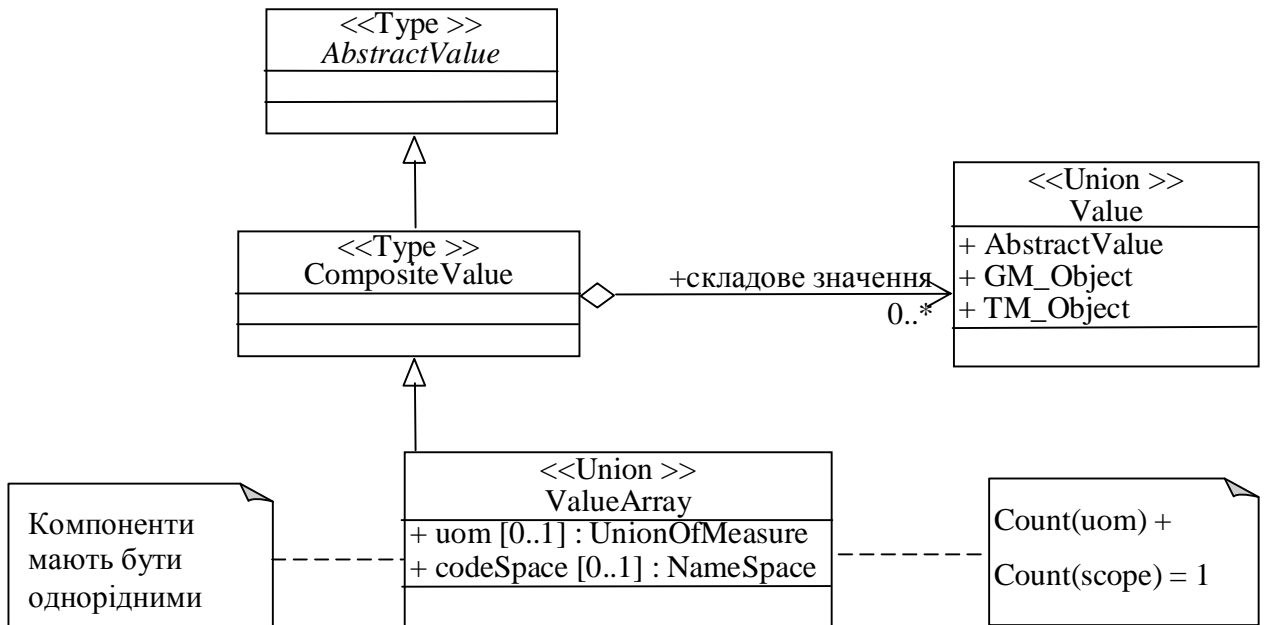


Рисунок Г.63 – Композитні значення

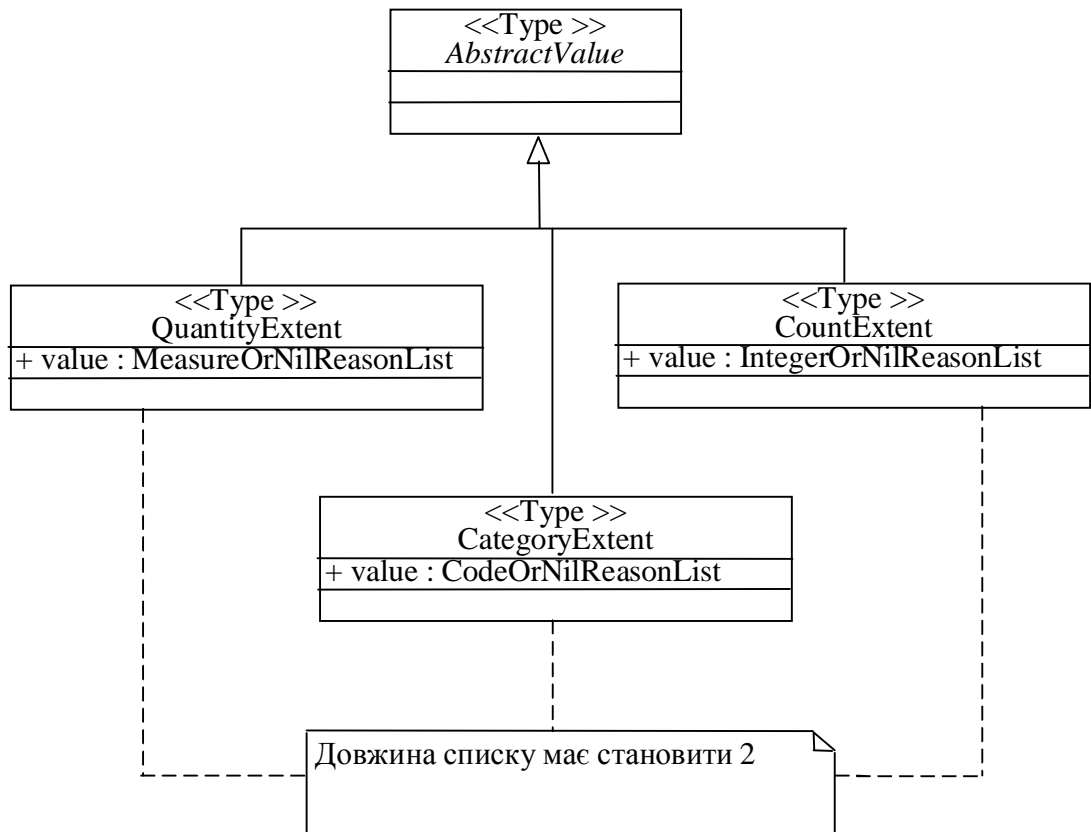


Рисунок Г.64 – Екстенти

### Г.3.16 Пакет “direction” (напрямок)

У цьому підрозділі визначені додаткові типи, використані в розділі 17 (див. рисунок Г.65). Ці типи використовують у схемі спостережень (див. додаток Г.3.17).

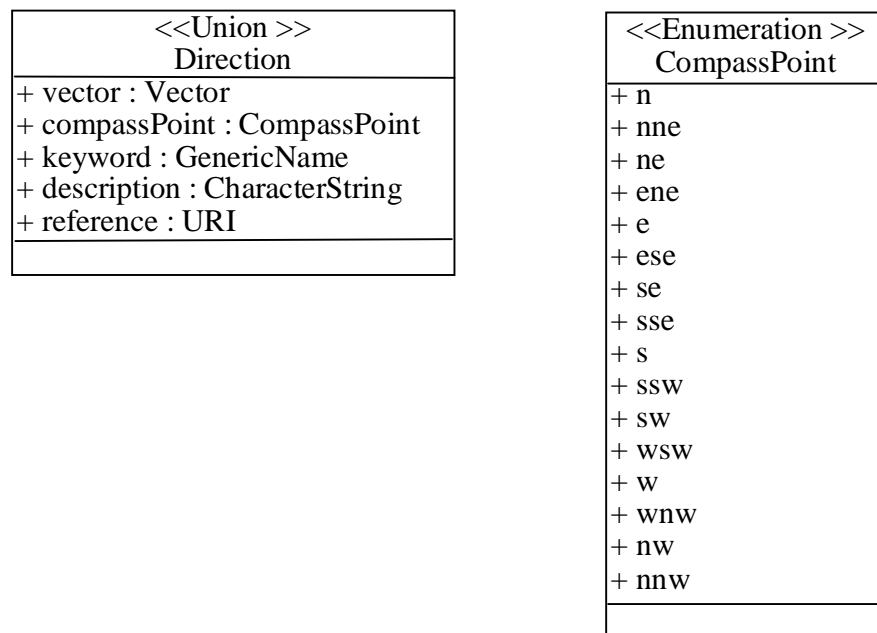


Рисунок Г.65 – Напрямок

### Г.3.17 Пакет “observation” (спостереження)

У цьому підрозділі визначені додаткові типи, використані в розділі 18 (див. рисунок Г.66).

Концепція спостереження додана до GML, оскільки концепцію спостережень вважають фундаментальною концепцією географічної інформації.

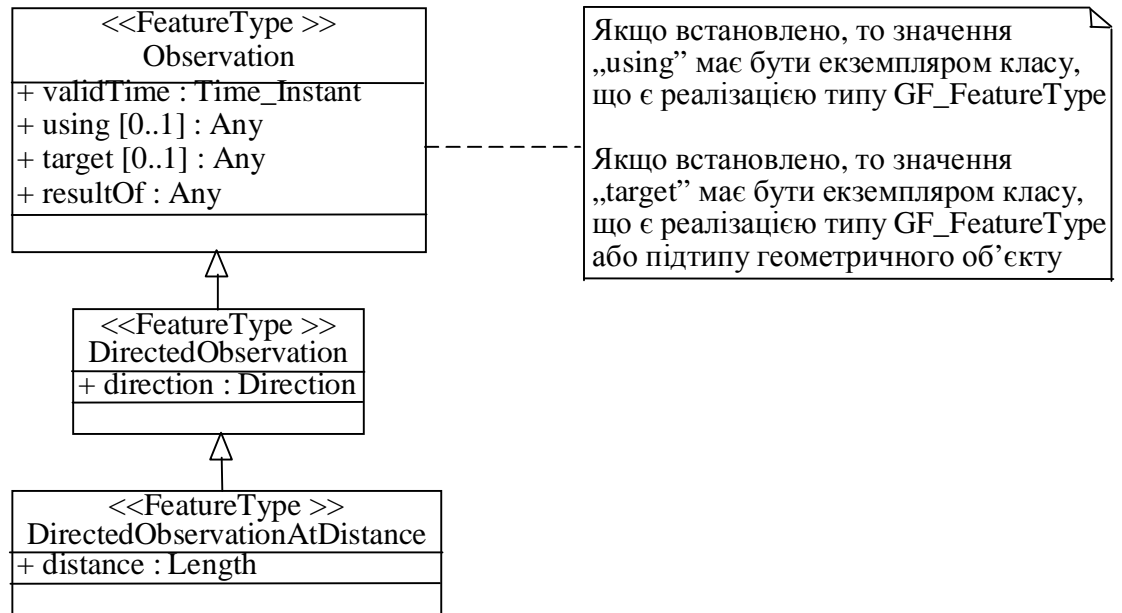


Рисунок Г.66 – Спостереження

#### ДОДАТОК Д (нормативний)

### ПРАВИЛА КОДУВАННЯ ПЕРЕХОДУ З UML ДО GML ПРИКЛАДНОЇ СХЕМИ

#### Д.1 Загальні концепції

Перехід від UML прикладної схеми, що узгоджена з ISO 19109, у відповідну GML прикладну схему базується на наборі правил кодування. Ці правила кодування відповідають правилам для GML прикладних схем та базуються на ISO 19118.

Правила отримують з правил для GML моделі та синтаксису, як описано в розділах з 7 по 21, зокрема в розділі 7. Правила кодування ISO 19118:2005 (додаток А) використовують завжди, коли це можливо та виправдано.

Правила, перелічені в цьому додатку, спрямовані на автоматичний перехід від UML прикладної схеми, яка відповідає ISO 19109 та ISO/TS 19103, до GML прикладної схеми (згідно з правилами, визначеними в розділі 21). Як результат цієї автоматизації, отримана GML прикладна схема не буде повністю використовувати можливості XML та XML схеми, але забезпечить XML реалізацію з добре визначеною та передбачуваною XML граматику, узгоджену з комплексом міжнародних стандартів ISO 19100.

Ці правила не вимагають, щоб усі GML прикладні схеми були вироблені з використанням цих правил. Всі схеми, що відповідають правилам, визначеним у розділі 21, є дійсними та відповідними GML прикладними схемами, незалежно від того, чи вони

виконані вручну, автоматично отримані з UML прикладної схеми або створені якимись іншими засобами.

Правила кодування схеми базовані на загальній ідеї про те, що визначення класів в прикладній схемі перекладають у декларації елементів та типів у XML схемі так, щоб об'єкти в моделі екземпляру могли перейти в відповідні структури елементів у XML документі.

## Д.2 Правила кодування

### Д.2.1 Загальні вимоги до кодування

#### Д.2.1.1 Прикладні схеми

##### Д.2.1.1.1 Загальні відомості (прикладна схема, пакети)

Для того, щоб бути дійсною вхідною інформацією при перекладі, UML прикладна схема повинна відповідати всім із нижчевикладених правил.

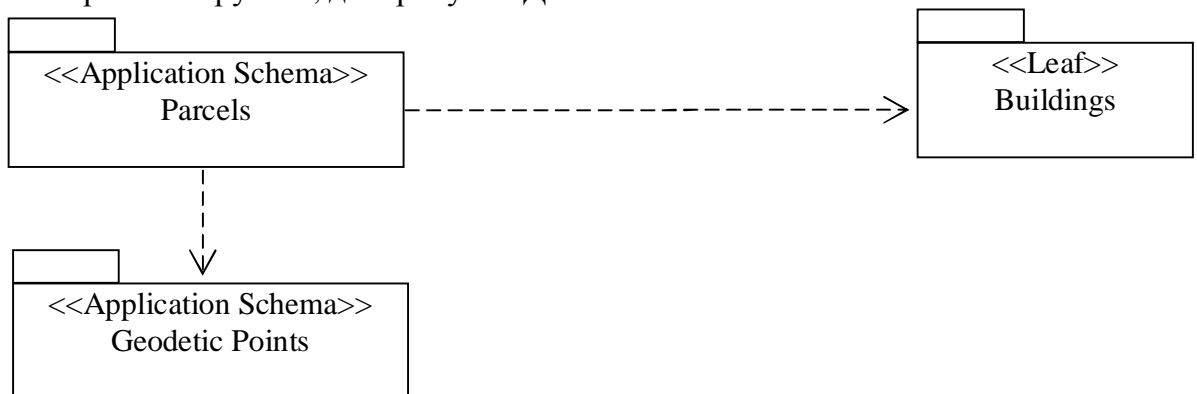
Дивіться додаткові вимоги в ISO 19118:2005, додаток А.2.1.

UML прикладна схеми має відповідати правилам, визначеним у ISO 19109 та ISO/TS 19103.

UML прикладну схему треба подавати пакетом із стереотипом <<Application Schema>>. Цей пакет має містити (прямо чи непрямо) всі елементи UML моделі, які треба перекласти в типи об'єктів у GML прикладній схемі. Цей пакет може включати інші пакети без стереотипу <<Application Schema>>, щоб групувати різні елементи UML моделі в прикладній схемі.

UML модель має бути повною та не може містити зовнішні посилання, крім виключень, які явним чином викладені нижче. Наперед визначені класи можна імпортувати із стандартизованих схем комплексу міжнародних стандартів ISO 19100. Класи з комплексу міжнародних стандартів ISO 19100, що реалізовані з допомогою GML схеми та використані UML прикладною схемою, треба визначати в пакеті з іменем "ISO19100" або будь-якому підпакеті пакету з цим іменем.

Залежності між пакетами треба моделювати явним чином. Елементи дозволу із стереотипами <<import>> або невизначені залежності між пакетами треба використовувати для подання залежності елементів в іншому пакеті. Всі інші елементи залежності треба ігнорувати, див. рисунок Д.1



**Рисунок Д.1** – Залежність між пакетами (довідковий)

Видимість усіх UML елементів треба встановити як "public". Тільки публічно видимі елементи мають бути частиною прикладних схем, які використовують для обміну даним між аплікаціями.

Документацію елементів у UML моделі треба зберігати в тегових значеннях “documentation”.

Унікальний XML простір імен треба асоціювати з UML прикладною схемою. Тегові значення “targetNamespace“ для цільового простору імен URI та “xmlns“ для скорочення треба встановлювати тоді і тільки тоді, коли пакетом подано UML прикладну схему. Номер версії пакету, яким подано UML прикладну схему, треба визначати в теговому значенні “version”, якщо це можна застосувати.

Номер версії пакету, яким подано UML прикладну схему, треба визначати в теговому значенні “version”, якщо воно застосовне.

Можна асоціювати GML профіль з прикладною схемою теговим значенням “gmlProfileSchema”. Якщо значення надано, воно має бути URL посиланням на розташування схеми GML профілю.

Якщо пакет треба перекласти в документ його власної XML схеми, треба використовувати тегове значення "xsdDocument" для забезпечення дійсного відносного імені файлу в документі схеми. Тегове значення треба встановлювати для кожного пакету, що репрезентує UML прикладну схему. Всі тегові значення "xsdDocument" у UML моделі мають бути унікальними.

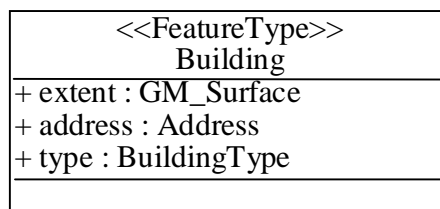
*Приклад 1.* Значення тегового значення "xsdDocument" могло б бути "GeodeticPoints.xsd" або "schemas/Parcels.xsd".

#### Д.2.1.1.2 Класи

Всі імена класів в одній і тій же прикладній схемі мають бути унікальними та "NCName", як визначено в W3C XML Namespaces:1999.

Типи географічних об’єктів треба моделювати як UML класи із стереотипом <<FeatureType>>.

**Примітка 1.** Треба відзначити, що ані в ISO 19109, ані в ISO 19118:2005 (додаток А) не розрізняють між типами географічних об’єктів та типами об’єктів: в ISO 19109 враховано тільки типи географічних об’єктів, тоді як в ISO 19118 (додаток А) подано класифікацію всіх типів географічних об’єктів та типів об’єктів. Однак, ця різниця є значущою у GML та часто потрібною на практиці в прикладних схемах. Різниця, розглянута в цьому додатку, є відповідним удосконаленням ISO 19118:2005 (додаток А).



**Рисунок Д.2** – Тип географічного об’єкту (довідковий)

Типи об’єктів треба моделювати як UML класи без стереотипу. Типи об’єктів та типи є типами, де екземпляри не мають ідентичності, але які не є типами географічних об’єктів<sup>10)</sup>.

<sup>10)</sup> Типи об’єктів не враховані явним чином у ISO 19109: вони зустрічаються тільки як типи даних типів властивості



*Приклад 1.* Приклади таких типів – геометрії, топології, референці системи. Екземпляри цих типів можуть мати, наприклад, ім'я та ідентифікатор.

UML класи із стереотипом <<Type>> можуть мати нуль чи більше операцій (вони не переходять у GML прикладну схему), атрибутів та асоціацій.

Стереотип <<Abstract>> не слід використовувати в прикладній схемі, тому що його використання може бути несумісним із використанням коректної UML нотації, таким чином спричиняючи непорозуміння.

Всі підтипи абстрактних типів, які можуть мати екземпляри, мають бути або типами географічних об'єктів, або типами об'єктів, або типами даних.

Переліки треба моделювати як UML класи із стереотипом <<Enumeration>>.

Списки кодів треба моделювати як UML класи із стереотипом <<CodeList>> (див. рисунок Д.3).

<<CodeList >> ParcelUsage (Використання ділянки)	
+ factory = 1	+ фабрика = 1
+ road = 2	+ дорога = 2
+ residential = 3	+ житлове = 3
+ offices = 4	+ офіси = 4
+ sea, river = 5	+ море, річка = 5
+ ...	+ ...

**Рисунок Д.3** – Списки кодів (довідковий)

Типи об'єднань треба моделювати як UML класи із стереотипом <<Union>> (як визначено в ISO 19107).

Всі інші типи даних треба моделювати як UML класи із стереотипом <<DataType>>.

<<DataType>> ParcelName
+ countryId : CharacterString
+ stateId : CharacterString
+ municipalityId : CharacterString
+ parcelIdPrefix : CharacterString
+ parcelIdSuffix: CharacterString

**Рисунок Д.4** – Типи даних (довідковий)

UML класи комплексу міжнародних стандартів ISO 19100, які є частиною GML профілю та для яких наданий базовий GML тип у таблиці Г.2 у стовпці "GML тип", можна поділити на підкласи в UML прикладній схемі. В підкласах можна додати додаткові властивості або можна перевизначити властивості підтипу з обмеженою множинністю чи доменом значень.

**Примітка 2.** Хоча перевизначення властивостей підтримується, ці перевизначені властивості будуть проігноровані в правилах перетворення, і аплікація повинна перевірити обмеження, реалізовані згідно з перевизначенням. Всі класи з іншими стереотипами, крім вищезгаданих, можуть бути частиною UML прикладної схеми, але будуть проігноровані.

**Примітка 3.** Коли прикладна схема посилається на типи, визначені іншими стандартами комплексу міжнародних стандартів ISO 19100 та реалізовані GML схемою, то імена класів мають відповідати одному з тих, що входять до списку у першому стовпці таблиці Г.2.

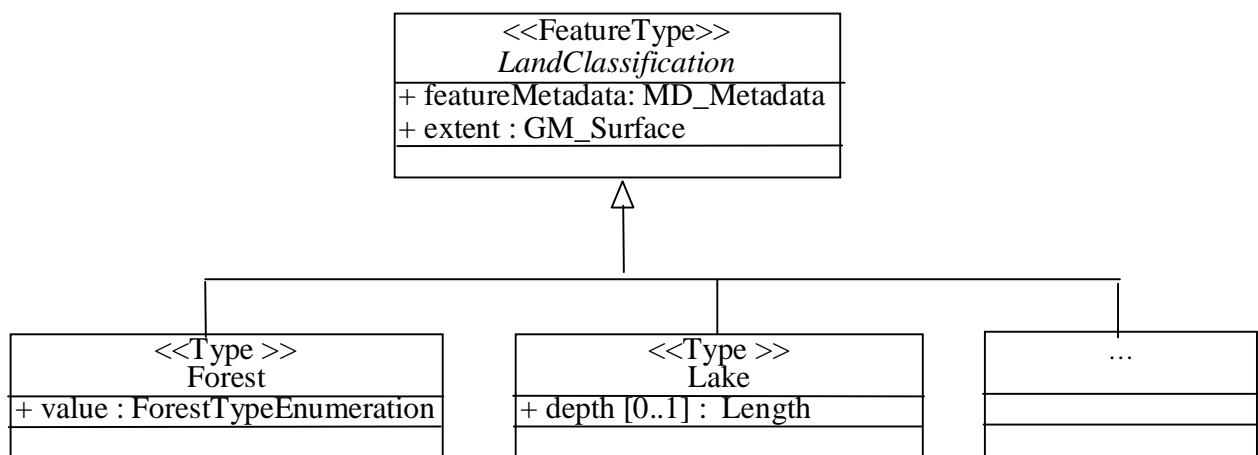
Відношення узагальнення можна визначити тільки між двома класами, що є або:

- обидва – типи географічних об'єктів,
- обидва – типи об'єктів, або
- обидва – типи даних.

Всі відношення узагальнення між класами не повинні мати стереотипу. Всі відношення узагальнення з іншими стереотипами будуть проігноровані. Властивість дискримінатора UML узагальнення має бути незаповненою.

Якщо клас є спеціалізацією іншого класу, то цей клас повинен мати тільки один надтип (множинне успадкування не підтримується).

Всі класи повинні мати стереотип, що визначає смисл цього класу. Класи без стереотипу розглядають як типи об'єктів, див. рисунок Д.5.



**Рисунок Д.5** – Відношення узагальнення між типами географічних об'єктів  
<довідкове>

### Е.2.1.1.3 Атрибути

Кожний UML атрибут абстрактного типу, типу географічних об'єктів, типу даних або типу об'єднання повинен мати ім'я і тип. Ім'я має бути "NCName", як визначено в просторах імен W3C XML Namespaces:1999. Якщо його множинність не є 1, то треба явним чином визначити множинність. Початкове значення можна визначити для атрибутів з числом, рядком чи типом переліку.

Тип має бути або наперед визначеним типом (див. Е.2.1.1.5), або класом, визначеним у UML моделі.

Кожний UML атрибут класу переліку повинен мати ім'я. Інформація типу залишається порожньою. Не можна приєднувати до атрибуту будь-яку інформацію про множинність, упорядкування чи початкове значення.

Кожний UML атрибут списку кодів повинні мати ім'я. Інформація про тип залишається порожньою. Не можна приєднувати до атрибуту будь-яку інформацію про множинність або упорядкування. Початкове значення можна визначити для документування коду для значення списку кодів. Якщо його пропущено, то значення (тобто, ім'я атрибуту) використовують як код.

Властивості UML класу невпорядковані. Для підтримки узгодженого упорядкування властивостей з UML моделі в перетворенні в XML схему треба визначити тежове значення “sequenceNumber” (домен значень: ціле число) для кожного атрибуту. Значення має бути унікальним для всіх атрибутів та кінців асоціацій класу.

#### Д.2.1.1.4 Асоціації та кінці асоціації

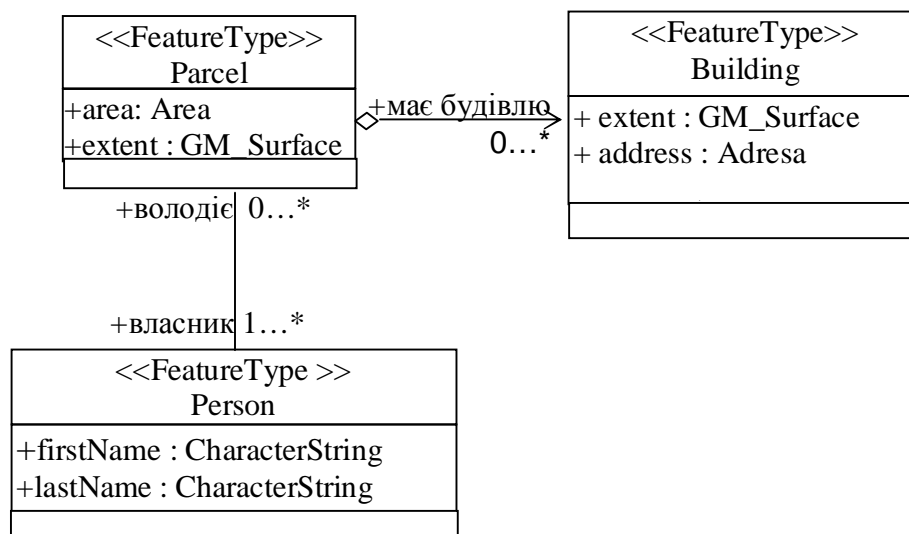
Кожна UML асоціація має бути асоціацією саме з двома кінцями асоціації. Обидва кінця асоціації мають з'єднуватися з **типом географічного об'єкту, об'єкту чи даних** та не можуть мати жодного стереотипу чи стереотипу <<association>> (інакше вся асоціація буде проігнорована).

Асоціація не повинна містити жодних властивостей.

Правила для кінців асоціації є такими:

- якщо кінець асоціації прохідний, його треба позначити як такий і дати йому ім'я ролі; кінець асоціації без імені треба ігнорувати, навіть якщо він позначений як прохідний; якщо ім'я надано, це має бути "NCName", як визначено просторами імен W3C XML Namespaces:1999;
- валентність треба подавати явним чином;
- вид агрегації треба визначати явним чином, якщо він не є “none” (*ніякий*);
- якщо цільовий клас кінця асоціації є типом даних, то видом агрегації має бути “composition” (композиція).

На рисунку Д.6 показано два приклади асоціації: одна асоціація прохідна у двох напрямках, а друга – агрегація, прохідна тільки в одному напрямку



**Рисунок Д.6** – Асоціації (довідковий)

Властивості UML класу невпорядковані. Для підтримки узгодженого впорядкування властивостей із UML моделі у перетворенні в XML схему треба визначити тежове значення “sequenceNumber” (домен значень: ціле число) для кожного кінця асоціації. Значення має бути унікальним для всіх атрибутів та кінців асоціацій класів.

**Д.2.1.1.5 Наперед визначені типи**

Наперед визначені типи із стандарту ISO/TS 19103, подані списком у Д.2.4.4, розглядають як “базові типи” у смислі ISO 19118:2005, додаток А (тобто, до них приєднане канонічне XML кодування).

**Д.2.1.1.6 OCL обмеження**

Всі OCL обмеження ігнорують. Оцінка дійсності моделі екземпляру стосовно цих обмежень є завданням аплікації, яка обробляє GML екземпляри.

**Примітка 1.** Мову Schematron можна використовувати для подання OCL обмежень як частини XML схеми, якою подано GML прикладну схему.

**Д.2.1.1.7 Інша інформація**

Всю іншу інформацію в UML прикладній схемі не використовують у правилах кодування та ігнорують.

**Д.2.1.2 Репертуар символів та мови**

Треба використовувати “UTF-8” or “UTF-16” як символне кодування файлів XML схеми (з асоційованим репертуаром символіки) відповідно до XML.

**Д.2.1.3 Обмінні метадані**

Обмінні метадані можна визначити для кожного географічного об’єкту чи колекції географічних об’єктів у документі GML екземпляру шляхом визначення в прикладній схемі елементів, модель контенту яких отримана з ”gml:AbstractMetadataPropertyType“, як визначено в пунктах 2.4.11 та Д.2.4.13.

До GML прикладної схеми не додано жодної спеціальної схеми для обмінних метаданих.

**Д.2.1.4 Ідентифікація набору даних та об’єкту**

Для ідентифікації об’єктів використовують унікальні ідентифікатори згідно з механізмом XML's ID.

**Примітка 1.** Механізм XML ID вимагає тільки, щоб ці ідентифікатори були унікальними ідентифікаторами в XML документі, де вони зустрічаються.

**Д.2.1.5 Механізм оновлення**

Не існує жодного явного механізму оновлення для географічних об’єктів, визначених у GML прикладній схемі. Припускають, що використовуються інші механізми для оновлення сховища даних.

**Примітка 1.** Прикладами є операція трансакції специфікації реалізації для OpenGIS® Web Feature Service/

**Д.2.2 Структура вхідних даних**

Див. опис структури вхідних даних в ISO 19118:2005, додаток А.3.

**Д.2.3 Структура даних на виході**

Це правило кодування базоване на XML рекомендації 1.0 та мові XML посилань (XLink), версія 1.0. Схема для структури даних на виході, яка управляє структурою обмінного формату, має бути дійсною XML схемою чи набором дійсних XML схем згідно з XML схемою 1.0 та правилами для прикладних схем (див. розділ 21).

Правила перетворення XML схеми визначені в наступному підрозділі.

**Д.2.4 Правила перетворення****Д.2.4.1 Загальні концепції**

Правила перетворення схеми визначають, як треба отримувати XML документи схеми (XSD) з прикладної схеми, поданої в UML відповідно до ISO 19109. Низка

загальних правил визначена в Д.2.4 для описування переходу від UML моделі, що відповідає інструкціям, описаним у підрозділі Д.2.1.

**Примітка 1.** У цьому додатку простір імен "xsd:" використовують для посилання на простір імен XML схеми, що є "<http://www.w3.org/2001/XMLSchema>". Простір імен "gml:" посилається на простір імен GML, який є "<http://www.opengis.net/gml/3.2>".

Правила базуються на GML моделі та синтаксисі, як описано в розділах 7-21 (особливо розділи 7, 9 та 21) та також на правилах кодування 19118:2005 (додаток А).

Правила кодування схеми базуються на загальній ідеї про те, що визначення класу в UML прикладній схемі переходять у декларації елементів та типів у XML схемі так, щоб об'єкти в моделі екземпляру можна було перекласти в відповідні структури елементів у XML документі.

Огляд цього переходу поданий у таблиці Д.1 :

**Таблиця Д.1** – Огляд кодування схеми

<b>Таблиця: UML® GML прикладна схема</b>	
<b>UML прикладна схема</b>	<b>GML прикладна схема</b>
Пакет	Один XML документ схеми на пакет (перехід за умовчанням)
<<Application Schema>> (прикладна схема)	Документ XML схеми
<<DataType>> (тип даних)	Глобальний елемент, модель контенту якого – XML complexType чи тип властивості глобального охоплення
<<Enumeration>> (перелік)	Обмеження xsd:string із значеннями переліку
<<CodeList>> (список кодів)	Об'єднання переліку та шаблону (перехід за умовчанням, альтернативний перехід як посилання на словник).
<<Union>> (об'єднання)	Група вибору, члени якої є GML об'єктами чи географічними об'єктами або об'єктами, які відповідають типам даних
<<FeatureType>> (тип об'єкту)	Глобальний елемент, модель контенту якого – XML тип глобального охоплення, отриманий шляхом прямого непрямого розширення абстрактного типу gml:AbstractFeatureType, типу властивості
Стереотипу немає або <<Type>>	Глобальний елемент, модель контенту якого – XML тип глобального охоплення, отриманий шляхом прямого чи непрямого розширення абстрактного типу gml:AbstractGMLType, типу властивості
Операції	Не заковані
Атрибут	Локальний xsd:element, тип – або тип властивості (якщо тип є комплексним типом), або простий тип
Роль в асоціації	Локальний xsd:element, тип – завжди тип властивості (тільки іменовані та проходимі ролі)
Загальні обмеження OCL	Не заковані

Примітка 1. <<FeatureType>> – це новий стереотип, який не зустрічається в ISO/TS 19103 чи ISO 19109 та використовується для зазначення того, що тип є реалізацією типу GF\_FeatureType та спеціалізацією від AbstractFeature.

Чисельність атрибутів та ролей в асоціаціях переходять в атрибути "minOccurs" та "maxOccurs" у деклараціях <xsd:element>. Докладний опис правил перекладу описано далі.

Для різних елементів UML моделі використовують різні тегові значення з тим, щоб контролювати перехід від UML до XML схеми. У таблиці Д.2 поданий список цих тегових значень.

**Таблиця Д.2 – Тегові значення**

Елемент UML моделі	Асоційовані тегові значення
<b>Пакет</b>	–documentation –xsdDocument –targetNamespace (only <<Application Schema>>) –xmlns (only <<Application Schema>>) –version (only <<Application Schema>>) –gmlProfileSchema (only <<Application Schema>>)
<b>Клас</b>	–documentation –noPropertyType –byValuePropertyType –isCollection –asDictionary (only <<CodeList>>) –xmlSchemaType (only <<Type>>)
<b>Атрибут та кінець асоціації</b>	–documentation –sequenceNumber –inlineOrByReference –sMetadata

#### Д.2.4.2 UML пакети

Створюють один документ XML схеми на пакет з теговим значенням "xsdDocument" з іменем файлу, визначеним цим теговим значенням.

Якщо тегове значення "xsdDocument" встановлене для пакету, то документ схеми містить всі компоненти XML схеми, що є результатами від UML класів, які належать цьому пакету. Якщо пакет не є UML прикладною схемою, то документ схеми треба включити в документ схеми, що містить компоненти схеми пакету, якому належить цей пакет.

Якщо тегове значення "xsdDocument" не встановлене для пакету, то всі компоненти схеми задекларовані в документі схеми, що містить компоненти схеми пакету, до якого належить цей пакет.

**Примітка 1.** Тегове значення обов'язкове для всіх пакетів із стереотипом <<Application Schema>>, але необов'язкове для інших пакетів.

Для кожного документу схеми треба встановлювати атрибути "targetNamespace" та "version" кореневого елемента згідно з теговими значеннями того ж імені в пакеті, яким подана UML прикладна схема, якій належать компоненти схеми в документі схеми; якщо тегове значення "version" не визначене, то треба використовувати значення

"unknown". Крім цього, треба визначити атрибут "xmlns" для цільового простору імен із значенням тегового значення "xmlns" як скороченням.

*Приклад 1.* "<http://www.myorg.com/myns>" може бути цільовим простором імен, а "myns" може бути асоційованим скороченням у документах схеми.

Залежності між пакетами треба використовувати для визначення потрібного імпорту інших схем та додаткових включень інших документів схеми:

- якщо компоненти схеми визначені цільовим пакетом відношень залежності, перебувають у тому ж самому цільовому просторі імен, що і такі ж із пакету постачальника, то документ схеми, що визначає компоненти схеми цільового пакету, є "включеним";
- в іншому разі документ схеми, яким подано пакет UML прикладної схеми, що містить цільовий акт, є "імпортованим".

*Приклад 2.* Перехід інформації (рисунок Д.1) може мати такий результат:

```
<?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>
```

#### Д.2.4.3 UML класи (загальні правила)

Визнаними стереотипами для UML класів є: відсутній стереотип, <<FeatureType>>, <<Type>>, <<DataType>>, <<Union>>, <<CodeList>> та <<Enumeration>>. Всі класи треба перекладати у відповідні категорії класів. Всі UML класи з іншими стереотипами будуть проігноровані.

Всі UML класи мають мати не більше одного надтипу.

Всі UML класи перекладають у поименовані типи. Суфікс "Type" додають до імені типу.

#### Д.2.4.4 UML класи (базові типи)

Базові типи з GML профілю ISO/TS 19103, подані списком у лівому стовпці таблиці Г.2 (починаючи з "CharacterString"), наперед визначені та можуть бути використані як тип даних атрибуту в прикладній схемі відповідно до ISO 19109. Визначено перехід у вбудований тип XML схеми ("xsd:") або GML ("gml:").

Якщо подано кілька імен у комірці таблиці, то ім'я, виділене грубим шрифтом, треба використовувати як тип за умовчанням у переході.

**Примітка 1.** Кілька значень у комірках правого стовпця використовують для підтримки також і зворотнього переходу в додатку Е.

*Приклад 1.* ISO/TS 19103 Integer (ціле число) переводять у "xsd:integer".

Якщо клас із стереотипом <<Type>> має канонічне XML кодування (тобто, з XML схеми), то ім'я типу XML схема відповідно до типу даних треба подавати як значення тегового значення "xmlSchemaType".

**Примітка 1.** У деяких випадках можна надавати перевагу канонічним кодуванням над структурними кодуваннями, що відповідають стандарту правил кодування з UML до GML, наприклад, якщо компактна структура, базована на простому контенті, уже добре відома в рамках прикладного домену.

#### Д.2.4.5 UML класи (типи даних)

UML класи з стереотипом <<DataType>> треба перекладати в комплексні типи XML схеми.

**Примітка 1.** Типи даних із іншими стереотипами, тобто <<Enumeration>>, <<CodeList>> та <<Union>>, та наперед визначені базові типи розглядають по-різному. Див. Д.2.4.4, Д.2.4.8, Д.2.4.9 та Д.2.4.10.

Якщо клас не має стереотипу, то це не є тип, отриманий у XML схемі; інакше він розширює його надтип, який не можна отримувати від gml:AbstractGMLType (прямо чи непрямо). Абстрактний надклас без будь-якого атрибуту або ролі в прохідних асоціаціях будуть проігноровані.

Для цих класів треба визначити глобальні XML елементи з відповідними установками для імені (ім'я UML класу), типу (ім'я UML класу плюс "Type"), абстрактності (якщо клас абстрактний) та груп заміни (кваліфіковане ім'я елемента надкласу gml:AbstractObject, якщо клас не має надкласу).

Іменованій комплексний тип треба створити для цих класів (має включати ім'я класу із суфіксом "PropertyType"), якщо клас не має тегового значення "noPropertyType" із значенням "true". Тип відповідає шаблону для властивостей асоціації, як визначено в GML (див. пункт 7.2.3), але без дозволу атрибутів Xlink.

*Приклад 1.* Тип даних "ParcelName" з рисунку Д.4 можна перекласти в:

```
<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>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
```

#### Д.2.4.6 UML класи (типи географічних об'єктів)

UML класи із стереотипами <<FeatureType>> походять прямо чи непрямо від gml:AbstractFeatureType. Якщо даний клас є класом без надтипу, він розширює безпосередньо gml:AbstractFeatureType; в інших випадках він розширює його надтип, який треба отримувати від gml:AbstractFeatureType (знову таки, прямо чи непрямо):



- для цих класів визначені глобальні XML елементи з відповідними установками для імені (ім'я UML класу), типу (ім'я UML класу плюс “Type”), абстрактності (істина, якщо клас абстрактний) та груп заміни (ім'я надкласу gml:AbstractFeature);
- якщо клас має тільки одну асоціацію, що є агрегацією чи композицією цільового класу, роль в асоціації перетворюють на елемент властивості, а клас має тежове значення "isCollection" із значенням "true" (істина), і додають групу атрибутів gml:AggregationAttributeGroup до комплексного типу географічного об'єкту;
- треба створювати іменованій комплексний тип для цих класів (має ім'я класу із суфіксом “PropertyType”), якщо клас не має тежового значення "noPropertyType" із значенням "true" (істина); тип відповідає шаблону для властивостей асоціації, як визначено в GML (див. пункт 7.2.3).
- треба створювати іменованій комплексний тип для цих класів (містить ім'я класу із суфіксом “PropertyByValueType”), якщо клас має тежове значення "byValuePropertyType" із значенням "true" (істина); тип є профілем шаблону для властивостей асоціації, як визначено в GML, обмеженим до форми “by value” (також див. пункт 7.2.3).

*Приклад 1.* “Building” з рисунку Д.2 можна перекласти таким чином:

```

<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"/>

```

#### **Д.2.4.7 UML класи (типи об'єктів)**

UML класи без стереотипу чи із стереотипом <<Type>> отримують прямим чи непрямым шляхом від gml:AbstractGMLType. Якщо даний клас є класом без стереотипу,

то він безпосередньо розширює `gml:AbstractGMLType`, а в інших випадках він розширює його надтип, який треба отримувати від `gml:AbstractGMLType` (також, прямо чи непрямо), але не від `gml:AbstractFeatureType` (також, прямо чи непрямо):

- для цих класів визначені глобальні елементи з відповідними установами імені (ім'я UML класу), типу (ім'я UML класу плюс "Type"), абстрактності (істина, якщо клас є абстрактним (ім'я надтипу чи "AbstractGML"));
- якщо клас має хоча б одну асоціацію, яка є агрегацією чи композицією цільового класу, то роль в асоціації перетворюють на елемент властивості, клас має тежове значення "isCollection" із значенням "true", і групу атрибутів `gml:AggregationAttributeGroup` додають до комплексного типу типу об'єкту;
- треба створювати іменованний комплексний тип для цих класів (містить ім'я класу із суфіксом "PropertyType"), якщо клас не має тежового значення "noPropertyType" із значенням "true" (істина); цей тип відповідає шаблону для властивостей асоціації, як визначено в GML (див. пункт 7.2.3);
- треба створювати іменованний комплексний тип для цих класів (містить ім'я класу із суфіксом "PropertyByValueType"), якщо клас має тежове значення "byValuePropertyType" із значенням "true" (істина); цей тип є профілем шаблону для властивостей асоціації, як визначено в GML, обмеженим формою "by value" (див. пункт 7.2.3).

*Приклад 1.*

```
<element name="Ellipse" type="ex:EllipseType" substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="EllipseType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <element name="center" type="gml:DirectPosition"/>
        <element name="semiminor" type="gml:VectorType"/>
        <element name="semimajor" type="gml:VectorType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

#### Д.2.4.8 UML класи (переліки)

UML класи із стереотипом <<Enumeration>> переходять у прості типи XML схеми. Базовим типом є "string", домен значень обмежений до множини літерних значень, як визначено іменами атрибутів UML класу.

*Приклад 1.*

```
<simpleType name="SignType">
  <restriction base="string">
    <enumeration value="-"/>
    <enumeration value="+"/>
  </restriction>
</simpleType>
```

**Д.2.4.9 UML класи (списки кодів)**

UML клас із стереотипом <<CodeList>> та без тегового значення "asDictionary" із значенням "true" (істина) треба перекладати подібно до переліку, але з такими відмінностями:

- facet "<pattern value='other: \w{2,}'/>" треба додати, що дозволить будь-які текстові значення в додаток до наперед визначених значень; ці довільні значення мають префікс "other: "
- якщо код визначений для значення списку кодів, то тільки код треба подавати як переліковий facet;
- закодоване значення треба кваліфікувати з appinfo анотацією з елементом gml:description, який визначає текстове значення в значенні переліку.

*Приклад 1.* Список кодів "ParcelUsage" із рисунку Д.3 можна подати таким чином:

```
<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>
    <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>

```

Якщо клас несе тегове значення "asDictionary" із значенням "true", треба використовувати gml:Dictionary для подання списку кодів.

*Приклад 2.* Список кодів "ParcelUsage" із рисунку Д.3 можна подати в словнику як:

```

<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#ParcelUsage">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#ParcelUsage">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#ParcelUsage">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#ParcelUsage">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#ParcelUsage">5</gml:identifier>
        </gml:Definition>
      </gml:dictionaryEntry>
    </gml:Dictionary>
  </gml:dictionaryEntry>
</gml:Dictionary>

```

У документі екземпляру в такому випадку посилання було б закодоване (з використанням `gml:CodeType` як моделі контенту, див. Д.2.4.11) таким чином:

```
<usage codeSpace="http://www.someorg.de/example/cl.xml#ParcelUsage">1</usage>
```

Атрибут `codeSpace` указує на словник, а значення є іменем внесення у цей словник.

Спосіб кодування списку кодів у GML прикладній схемі також визначає, як треба кодувати елементи властивості, що мають списки кодів в якості домену їх значень; див. Д.2.4.11.

#### Д.2.4.10 UML класи (об'єднання)

UML класи із стереотипом `<<Union>>` перекладають як комплексні типи XML схеми. Ці класи перекладають подібно до типів даних (див. Д.2.4.5), але замість `<xsd:sequence>` властивостей використовують `<xsd:choice>` з тим, щоб визначити саме одну з властивостей в екземплярі об'єднання.

*Приклад 1.*

```
<complexType name="RemoteResourceType">
  <choice>
    <element name="name" type="string"/>
    <element name="uri" type="anyURI"/>
  </choice>
</complexType>
```

#### Д.2.4.11 UML атрибути та ролі в асоціації

UML атрибут або роль в асоціації об'єкту або типу географічного об'єкту переходять у локальний елемент з тим же іменем у комплексному типі, що визначає модель контенту об'єкту чи типу географічного об'єкту. Атрибути `minOccurs` та `maxOccurs` установлюють відповідно до визначень у UML моделі (докладніше про перехід див. ISO 19118:2005, додаток А). Тип залежить від типу значення властивості в UML.

Якщо тип значення властивості має простий контент, то такий тип використовують безпосередньо.

*Приклад 1.* `<element name="count" type="integer"/>`

Якщо тип значення властивості має комплексний контент, то треба використовувати тип властивості. Кодування типу властивості за умовчанням дозволяє як включене подання для типів об'єктів та географічних об'єктів, так і їх подання шляхом посилань, а також включене подання для типів даних та об'єднань. Для типів об'єктів та географічних об'єктів подання можна обмежити до включеного або шляхом посилань, використовуючи тегові значення `"inlineOrByReference"` із значенням `"inline"` або `"byReference"` відповідно. Якщо тегове значення відсутнє або його значення становить `"inlineOrByReference"`, то треба використовувати кодування за умовчанням.

Якщо атрибут або роль в асоціації є властивістю метаданих, то тип властивості має розширювати `gml:AbstractMetadataPropertyType` (див. підпункт 7.2.6); властивість метаданих є властивістю з теговим значенням `"isMetadata"` із значенням `"true"` (істина) або з такою властивістю, значенням якої є клас, визначений в 19115:2003. Якщо роллю в асоціації є цільовий кінець агрегації або композиції, то тип властивості має розширювати `gml:AbstractMemberType` (див. підпункт 7.2.5.1), коли це не є властивістю метаданих.

Якщо роль в асоціації є цільовим кінцем композиції атрибуту об'єкту-значення, то елемент властивості має додавати Schematron обмеження, яке гарантує, що атрибут owns групи gml:OwnershipAttributeGroup є "true". Schematron обмеження має відповідати такому шаблону:

```
<sch:pattern name="composition">
  <sch:rule context="qualified property name of the object element">
    <sch:report test=" qualified property name/@owns='true'"> Ця властивість є
композицією, значення мають бути у власності</sch:report>
  </sch:rule>
</sch:pattern>
```

*Приклад 2.* Для властивості ex:representativeLocation типу географічного об'єкту ex:МуFeature, яка контролює точковий об'єкт, що описує позицію, це можна описати таким чином:

```
<element name="representativeLocation" type="gml:PointPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern name="composition">
        <sch:rule context="ex:representativeLocation">
          <sch:report test="@owns='true'"> Ця властивість є композицією,
значення мають бути у власності</sch:report>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

Якщо тип властивості уже визначений у прикладній схемі як іменованний тип (це можна визначити шляхом перевірки тегових значень "noPropertyType" та "byValuePropertyType"), то треба посилатися на цей компонент схеми; в інших випадках треба визначити анонімну властивість локально в елементі властивості.

Якщо закодована властивість є кінцем асоціації, то інший кінець асоціації також треба кодувати у GML прикладній схемі, ім'я властивості іншої асоціації треба кодувати в елементі gml:reversePropertyName в appinfo нотації елемента властивості (див. підпункт 7.2.3.9).

*Приклад 3.* Подання шляхом посилання чи включення:

```
<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>

```

або

```

<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>

```

Альтернативно тип властивості може підтримувати тільки одне з подань, включене або шляхом посилання, залежно від тегового значення "inlineOrByReference".

*Приклад 4.* Подання тільки шляхом включення:

```

<element name="owner" type="ex:PersonPropertyByValueType" minOccurs="0"/>
...
<complexType name="PersonPropertyByValueType">
  <sequence>
    <element ref="ex:Person"/>
  </sequence>
</complexType>

```

або

```

<element name="owner" minOccurs="0">
  <complexType>
    <sequence>
      <element ref="ex:Person"/>
    </sequence>
  </complexType>
</element>

```

Якщо підтримується подання тільки шляхом посилання, то елемент властивості треба кваліфікувати з елементом appinfo анотації gml:targetElement, що визначає ім'я кваліфікованого елемента цільового типу.

```

<element name="targetElement" type="string"/>

```

Якщо закодована властивість є кінцем асоціації, та інший кінець асоціації також закодований у GML прикладній схемі, то ім'я властивості іншого кінця асоціації треба

закодувати в іншому елементі appinfo анотації gml:reversePropertyName, визначеному вище.

*Приклад 5.* Подання тільки шляхом посилання:

```
<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>
```

Залежно від кодування класу, UML атрибут списку кодів або переліку перекладають в елемент або з рядковим значенням (домен значень: значення переліку або список кодів) або із значенням, що посилається на відповідний запис у словнику. В екземплярі можна посилатись явним чином на словник із використанням атрибуту codeSpace. Значення за умовчанням для URI, яким поданий словник, можна надати з використанням елементу appinfo анотації gml:defaultCodeSpace.

```
<element name="defaultCodeSpace" type="anyURI"/>
```

*Приклад 6.* Список кодів “BuildingType” можна подати таким чином:

```
<element name="type" type="ex:BuildingTypeType"/>
```

або

```
<element name="type" type="gml:CodeType">
  <annotation>
    <appinfo>
      <gml:defaultCodeSpace>http://www.someorg.de/example/cl.xml#BuildingType</gml:defaultCodeSpace>
    </appinfo>
  </annotation>
</element>
```

Якщо UML атрибут або роль в UML асоціації перевизначена (тобто, підклас містить атрибут чи роль в асоціації з тим же іменем, що і в надтипі), то ця властивість не є частиною моделі контенту підтипу. Розробник аплікації відповідає за гарантії відповідності екземплярів таким обмеженням, поданим у концептуальній моделі.

Всі атрибути та ролі в асоціації треба перетворювати в порядку зростання тегового значення “sequenceNumber”.

#### **Д.2.4.12 Документація**

Тегові значення “documentation” з елементів UML моделі перекладають в елементи анотації чи документації в файлах XML схеми.

*Приклад 1.*

```
<element name="curveProperty" type="gml:CurvePropertyType">
  <annotation>
    <documentation>Цей елемент властивості містить елемент кривої чи посилається на криву через XLink атрибути. curveProperty є наперед визначеною властивістю, яку можуть
```



використовувати GML прикладні схеми завжди, коли GML географічний об'єкт має властивість із значенням, яким можна замінити AbstractCurve.</documentation>

</annotation>

</element>

#### **Д.2.4.13 Імпортовані класи з комплексу міжнародних стандартів ISO 19100**

У додаток до правил, визначених вище, при імпорті класів з комплексу міжнародних стандартів ISO 19100 UML прикладною схемою треба застосовувати такі правила:

– Треба визнавати класи з комплексу міжнародних стандартів ISO 19100, що реалізовані GML схемою. Використання класів з комплексу міжнародних стандартів ISO 19100 має відповідати ISO 19109. Перехід доречних класів з комплексу міжнародних стандартів ISO показаний у таблиці Г.2.

– Якщо клас, узятий з ISO 19115:2003 та реалізований в ISO/TS 19139, використовувати як тип властивості, то треба визначити тип анонімної властивості, що розширює gml:AbstractMetadataPropertyType. Інкапсульований елемент об'єкту є відповідним елементом об'єкту для типу метаданих, як визначено в ISO/TS 19139.

#### **Д.2.4.12 Класи, імпортовані з інших концептуальних моделей із наперед визначеним XML кодуванням**

У додаток до правил, визначених вище, при імпорті класів UML прикладною схемою класів з іншої UML моделі, для якої вже визначено стандартне XML кодування, застосовують наступні правила.

– Треба визначити розширення до таблиці Г.2. Таблицю треба розповсюджувати разом із прикладною схемою в UML.

– Перехід доречних класів від імпортованої моделі до XML схеми нормативно визначено в цій таблиці.

## Д.3 Приклад &lt;довідковий&gt;

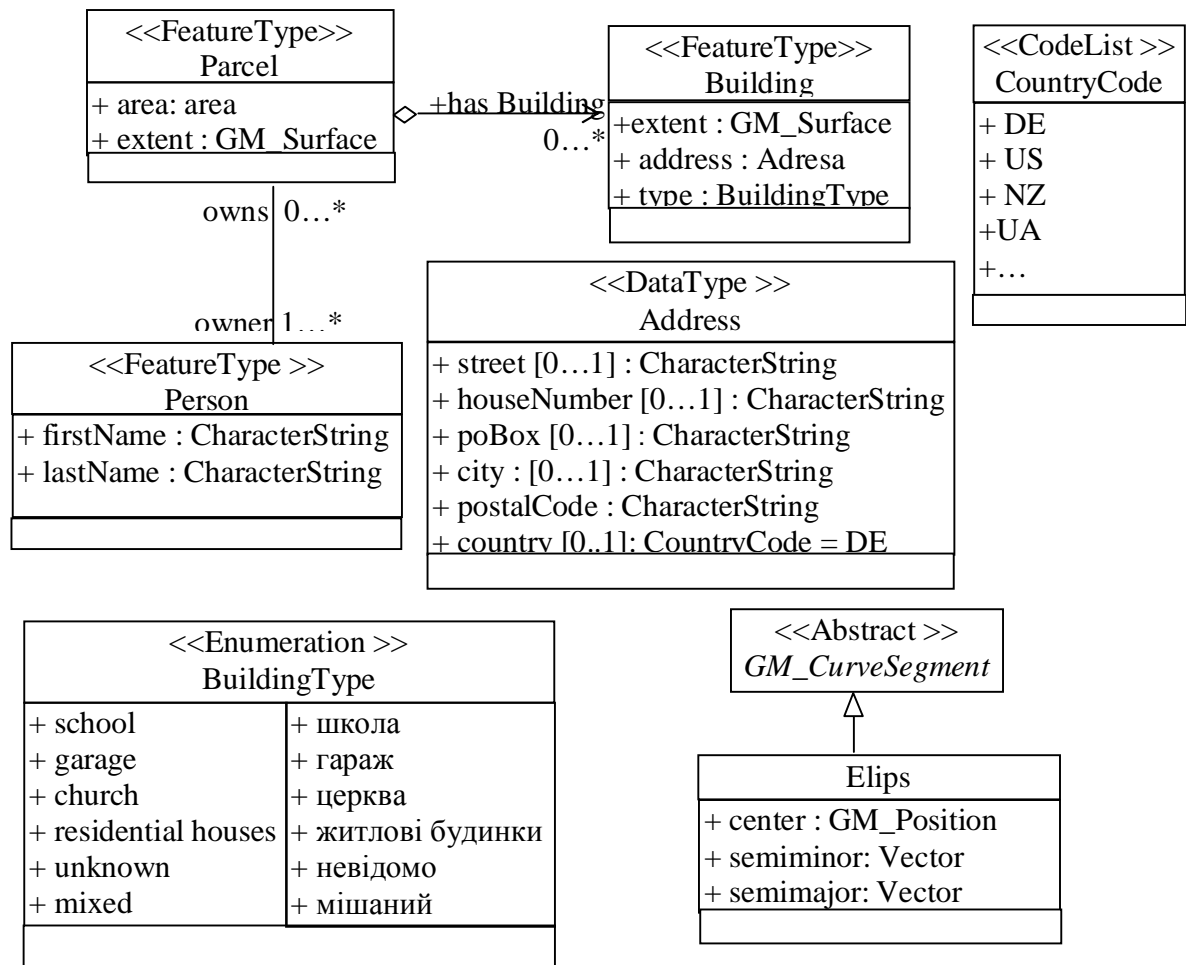


Рисунок Д.7 – Приклад прикладної схеми

Прикладну схему, показану на рис. Д.7, можна закодувати таким чином:

```

<?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>ex: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>
    </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="houenumber" 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>
          <element name="center" type="gml:DirectPositionType"/>
          <element name="semiminor" type="gml:VectorType"/>
          <element name="semimajor" type="gml:VectorType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>

```

ДОДАТОК Е  
(нормативний)

**ПРАВИЛА КОДУВАННЯ ПЕРЕХОДУ ВІД GML ДО UML ПРИКЛАДНОЇ  
СХЕМИ**

**Е.1 Загальні концепції**

Перехід від GML прикладної схеми до прикладної схеми в UML, яка відповідає ISO 19109, базується на наборі правил кодування. Ці правила кодування узгоджені з правилами для GML прикладних схем, як описано в розділах з 7 по 21, особливо в розділах 7, 9 та 21.

Правила, перераховані в підрозділі Е.2, націлені на автоматичний перехід від GML прикладної схеми до UML прикладної схеми, узгодженої з ISO 19109 та ISO/TS 19103.

Ці правила не вимагають, щоб усі GML прикладні схеми були генеровані для виконання вимог, задокументованих у цьому додатку. Всі схеми, які відповідають правилам, визначеним у розділі 21, є дійсними та узгодженими GML прикладними схемами.

Цей додаток треба використовувати, якщо є вимога у прикладному домені отримати прикладну схему на UML, узгоджену з ISO 19109, з GML прикладної схеми.

Скорочення XML простору імен "xsd" використовують для посилання на простір імен XML схеми, яким є "<http://www.w3.org/2001/XMLSchema>".

Скорочення XML простору імен “gml” посилається на XML простір імен GML, яким є “<http://www.opengis.net/gml/3.2>”.

Крім того, GML імпортує визначення з таких просторів імен:

- скорочення XML простору імен “xlink”, що посилається на XML простір імен для посилань, яким є <http://www.w3.org/1999/xlink>;
- термін “GML простори імен”, що використовують далі для посилання на простори імен “gml” and “xlink”.

## **Е.2 Правила кодування**

### **Е.2.1 Загальні вимоги до кодування**

#### **Е.2.1.1 Загальні зауваження**

Правила кодування схеми базуються на загальній ідеї про те, що відповідні декларації типів та елементів у XML схемі перекладають у визначення класів у UML прикладній схемі, тому ці структури елементів у XML документі можна перекласти в об’єкти в моделі екземпляру.

#### **Е.2.1.2 GML схема**

##### **Е.2.1.2.1 Загальний огляд**

Для того, щоб бути дійсним вхідним матеріалом для переходу, GML прикладна схема має відповідати вимогам доречних класів відповідності у підрозділі 2.2, щонайменше, “всі GML прикладні схеми”, “GML прикладна схема, яку треба перетворити в прикладні схеми в UML ISO 19109” та “GML прикладні схеми, що визначають географічні об’єкти та колекції географічних об’єктів”.

GML прикладна схема повинна мати та містити визначення тільки для цільового простору імен.

GML прикладна схема може імпортувати визначення з XML просторів імен, відмінних від її цільового простору імен.

GML прикладна схема складається з набору одного чи більше документів XML схеми, таким чином, що:

- всі документи мають унікальні імена;
- документи містять елементи `xsd:include` для інших документів схеми з тим же самим цільовим простором імен;
- один документ схеми вищого рівня для цільового простору імен GML прикладної схеми не включений іншими документами схеми для цільового простору імен, але прямо чи непрямо включає всі інші документи схеми для цільового простору імен, якщо такі є;
- документи схеми містять елементи `xsd:import` для XML просторів імен, відмінних від цільового простору імен, та для документів схеми, що містять визначення в цих XML просторах імен;
- всі включені та імпортовані документи схеми, доступні через URI, визначені атрибутом `schemaLocation` на елементах `xsd:include` або `xsd:import`, що посилаються на них;
- перевірючий XML парсер вирішує всі залежності серед визначень, що містяться в наборі документів схеми;
- перевірючий XML парсер перевіряє набір документів схеми без помилок;

– перевірючий XML парсер перевіряє документ XML екземпляру, що містить елементи та атрибути, якими подають всі визначення з цільового простору імен GML прикладної схеми без помилок.

Документування визначень, які містяться в GML прикладній схемі, треба зберігати у вкладених елементах `xsd:annotation` та `xsd:documentation` у елементах визначення схеми.

Версія GML прикладної схеми, якщо вона застосовна, має міститися в атрибуті версії елемента `xsd:schema` із схеми вищого рівня для її цільового простору імен.

Імена всіх глобальних типів та елементів у GML прикладній схемі мають бути унікальними.

GML прикладна схема не має визначати будь-які елементи з анонімними типами для об'єктів.

GML прикладна схема не має визначати будь-які XML атрибути чи іменовані групи.

Кожний комплексний тип у GML прикладній схемі має бути або GML типом об'єкту, або GML типом географічного об'єкту, або, GML типом даних або GML типом властивості.

Комплексні типи із простим контентом не мають бути визначені в GML прикладній схемі.

Ім'я всіх типів у GML прикладній схемі має закінчуватися з суфіксом "Type".

Суфікс "RestrictionType" в імені комплексного типу можна використовувати тільки для абстрактного типу, отриманого шляхом обмеження, та який є базовим типом саме одного комплексного типу, що походить від цього типу шляхом розширення та має те ж саме ім'я, що і обмежений тип, за виключенням того, що "RestrictionType" замінено на "Type".

Суфікс "PropertyType" в імені комплексного типу треба використовувати тільки для типу, що може мати екземпляри та відповідає шаблону для типів GML властивості шляхом посилання або значення. Має існувати комплексний тип (GML тип об'єкту або GML тип географічного об'єкту) з тим же іменем, що і "PropertyType", замінений на "Type".

Суфікс "PropertyValueType" в імені комплексного типу можна використовувати тільки для типу, що може мати екземпляри та відповідає шаблону для типів властивості через значення у GML. Має існувати комплексний тип (GML тип даних, GML тип об'єкту або GML тип географічного об'єкту) з таким же іменем, що має "PropertyValueType", замінений на "Type".

**Примітка 1.** Ці правила жорстко обмежують можливі форми GML прикладних схем.

#### **E.2.1.2.2 GML типи об'єктів, включно з GML типами географічних об'єктів**

Кожен GML тип об'єкту, визначений у GML прикладній схемі, повинен мати модель контенту, що прямо чи непрямо походить від `gml:AbstractGMLType` та повинна мати атрибут `gml:id`.

Кожен GML тип об'єкту особливого виду, визначений у GML прикладній схемі, має походити від найбільш спеціалізованого GML типу об'єкту з простору імен "<http://www.opengis.net/gml/3.2>" подібного виду (з сумісною семантикою), що можливо було б використовувати для визначення його моделі контенту. Тому GML типи об'єкту, визначені в GML прикладній схемі для подання географічних об'єктів (GML типи

географічних об'єктів), мають походити від `gml:AbstractFeatureType` замість `gml:AbstractGMLType`; GML типи об'єктів, визначені в GML прикладній схемі для подання геометричних точок, мають походити від `gml:PointType` замість `gml:AbstractGeometryType`, і т. і.

GML типи об'єктів, визначені в GML прикладній схемі, що походять від GML типів об'єктів за межами цільового простору імен, мають походити безпосередньо від одного з GML типів об'єкту, перерахованих у третьому стовпці таблиці Г.2, де в першому стовпці у тому ж самому рядку надано ім'я класу, визначеного комплексом міжнародних стандартів ISO 19100 або абстрактними типами `gml:AbstractGMLType` чи `gml:AbstractFeatureType`.

Визначення схеми абстрактних GML типів об'єкту має містити атрибут “abstract” із значенням “true”.

Ім'я абстрактних GML типів об'єктів має починатися з префіксу “Abstract”.

Визначення в схемі GML типів об'єкту, для яких не можна визначити підтипи, має містити атрибут “final” із значенням “всі” (“all”).

Властивості GML типу об'єкту треба визначити в елементі `xsd:sequence`.

#### **Е.2.1.2.3 Глобальні елементи для типів об'єкту GML**

Один глобальний елемент треба визначити для кожного типу GML об'єкту в GML прикладній схемі.

Іменем цього елементу має бути ім'я GML типу об'єкту без “Type”-суфіксу.

Елемент повинен мати атрибут `substitutionGroup`, значенням якого є ім'я глобального XML елементу, типом якого є базовий тип GML типу об'єкту.

#### **Е.2.1.2.4 Типи властивості за умовчанням для типів GML об'єктів**

GML тип властивості за умовчанням можна визначити в GML прикладній схемі для кожного GML типу об'єкту, визначеного в тій GML прикладній схемі.

GML тип властивості має використовувати або прямо чи непрямо успадковувати один із типів властивості, які визначені у пункті 7.2.3, або має бути визначений відповідно до шаблонів, визначених у цьому підрозділі.

Іменем цього типу властивості має бути ім'я типу GML об'єкту з суфіксом “Type”, заміненим на “PropertyType”.

Якщо жодний тип властивості за умовчанням не визначений для типу GML об'єкту, у прикладній схемі треба використовувати `gml:ReferenceType` як тип властивості за умовчанням GML типу об'єкту.

#### **Е.2.1.2.5 Включені типи властивості для GML типів об'єктів**

GML тип властивості за умовчанням для включених властивостей можна визначити в GML прикладній схемі для кожного GML типу об'єкту, визначеного в тій GML прикладній схемі.

GML тип властивості має або бути успадкованим прямо чи непрямо від `gml:InlinePropertyType`, або він має бути визначений згідно із шаблонами, визначеними в підпункті 7.2.3.8. Використання групи `gml:AssociationAttributeGroup` у таких властивостях заборонено.

Іменем цього типу властивості має бути ім'я GML об'єкту з суфіксом “Type”, заміненим на “PropertyByValueType”.

Якщо не визначено жодного типу властивості за умовчанням для включених властивостей для GML типу об'єкту, прикладна схема має використовувати



`gml:AssociationRoleType` як тип властивості за умовчанням для включених властивостей GML типу об'єкту.

#### **E.2.1.2.6 GML типи даних, включно з GML типами об'єднань**

Комплексний тип, визначений у GML прикладній схемі, що не походить прямо чи непрямо від `gml:AbstractGMLType`, називають GML типом даних.

Властивості GML типу даних мають приймати одну з таких форм:

- властивості комплексного типу, як і властивості всіх його базових типів, визначені в елементі `xsd:sequence` із значеннями `minOccurs` та `maxOccurs`, рівними 1;
- якщо GML тип даних не походить від будь-якого базового типу, то властивості можна визначити або одним елементом `xsd:sequence` із значеннями `minOccurs` та `maxOccurs`, рівними 1, або одним елементом `xsd:choice` із значеннями `minOccurs` та `maxOccurs`, рівними 1.

Модель контенту комплексного типу не повинна включати атрибут `gml:id`.

#### **E.2.1.2.7 Типи властивостей за умовчанням для GML типів даних**

GML тип властивості за умовчанням можна визначити в GML прикладній схемі для кожного GML типу даних, визначеного в цій GML прикладній схемі.

GML тип властивості має або бути успадкованими прямо чи непрямо від `gml:InlinePropertyType`, або його треба визначити відповідно до шаблонів, визначених у підпункті 7.2.3.8. Використання групи `gml:AssociationAttributeGroup` в таких властивостях заборонено.

Іменем цього типу властивості має бути ім'я GML типу даних з суфіксом "Type", замінене на "PropertyValueType".

Якщо не визначено жодного типу властивості за умовчанням для включених властивостей для GML типу даних, то у прикладній схемі треба використовувати `gml:AssociationRoleType` як тип властивості за умовчанням для включених властивостей GML типу даних.

#### **E.2.1.2.8 Переліки**

Простий тип, визначений у GML прикладній схемі, що є обмеженням, накладеним на `xsd:string` із використанням тільки `xsd:enumeration facet`, називають переліком.

#### **E.2.1.2.9 Списки кодів**

Простий тип, визначений у GML прикладній схемі, що є об'єднанням переліку та простого типу, тобто обмеженням, накладеним на `xsd:string` із використанням тільки одного `xsd:pattern facet` із значенням "other: \w{2,}", називають списком кодів.

Значення переліку можна кваліфікувати з допомогою `appInfo` анотації (елемент `gml:codeListValue`), що визначає, що значенням переліку є значенням коду іншого значення переліку; асоційоване значення переліку подають як текстове значення елементу `gml:codeListValue`.

#### **E.2.1.2.10 Глобальні елементи для GML типів даних, переліків та списків кодів**

Не можна визначати жодного глобального XML елемента для переліків або списків кодів, визначених у GML прикладній схемі.

#### **E.2.1.2.11 Наперед визначені базові типи**

Прості типи з XML прикладної схеми та GML простору імен, перелічені в четвертому стовпці у таблиці Г.2, можна використовувати в GML прикладній схемі.

Жодні інші прості типи з цих просторів імен не можна використовувати в GML прикладній схемі.

#### **E.2.1.2.12 GML властивості**

Кожну властивість GML типу об'єкту чи географічного об'єкту (за виключенням властивостей, визначених у GML просторі імен) або GML типу даних чи типу об'єднання треба подавати одиночним локально визначеним елементом `xsd:element`. Локально визначений означає, що ім'я та тип елемента треба подавати явним чином в декларації елемента (не має бути жодних посилань на глобальні XML елементи). Елемент може містити значення `minOccurs` та `maxOccurs`. Іменем цього елемента має бути ім'я властивості, а тип має бути або простим типом, або типом властивості.

#### **E.2.1.2.13 Schematron обмеження**

Всі Schematron обмеження ігнорують.

#### **E.2.1.2.14 Імпортовані елементи та типи з інших XML просторів імен**

Якщо інші XML компоненти схеми імпортують з інших просторів імен, відмінних від тих, які зустрічаються в XML схемі чи GML, треба визначити доречні записи як розширення до таблиці Г.2.

#### **E.2.1.2.15 Інша інформація**

Всю іншу інформацію в GML прикладній схемі у правилах кодування не використовують та ігнорують.

#### **E.2.1.3 Репертуар символів та мови**

Символьне кодування, яке використовують для схем, визначає наявний репертуар символів.

#### **E.2.1.4 Обмінні метадані**

Обмінні метадані можна визначити для кожного географічного об'єкту чи колекції географічних об'єктів у документі GML екземпляру<sup>11)</sup>.

Не було додано жодної спеціальної схеми для обмінних метаданих до GML прикладної схеми.

#### **E.2.1.5 Ідентифікація наборів даних та об'єктів**

Унікальні ідентифікатори `gml:id`, згідно до підпункту 7.2.4.5 та механізму XML's ID, треба використовувати для ідентифікації GML об'єктів.

#### **E.2.1.6 Механізм оновлення**

Не слід визначати жодний механізм оновлення для типів географічних об'єктів, визначених у GML прикладній схемі. Припускають, що для оновлення сховища даних моделі екземпляру використовують інші механізми.

#### **E.2.1.7 Структура вхідних даних**

Схема для структури вхідних даних визначена XML схемою 1.0, частина 1: Структури та частиною 2: Рекомендації W3C по типах даних та Правилами для GML прикладних схем (див. розділ 21).

#### **E.2.2 Структура даних на виході**

Див. докладніше опис структури даних на виході в ISO 19118:2005, підрозділ A.3.

#### **E.2.3 Правила перетворення**

##### **E.2.3.1 Загальні концепції**

<sup>11)</sup> шляхом використання елементів властивості, модель контенту походить від `gml:AbstractMetadataPropertyType`, та, наприклад, ISO/TS 19139 XML схеми кодування ISO 19115:2003.

Правила перетворення схеми, визначені в наступних підрозділах, описують перехід від GML прикладної схеми, що відповідає інструкціям, описаним у підрозділі E.2.1, в UML прикладну схему, яка відповідає правилам, визначеним у ISO 19109 та ISO/TS 19103 із використанням правил кодування стандарту ISO 19118:2005, додаток А та, зокрема, загальний моделі екземплярів, описаній у підрозділі А.3. Ці правила також базуються на поточних правилах для GML моделі та синтаксису, як описано в розділах з 7 по 21 (особливо в розділі 7).

Правила перетворення схеми визначають перехід визначення з (набору) дійсних документів GML прикладної схеми (XSDs) до набору UML пакетів. Пакет вищого рівня із стереотипом <<Application Schema>> створюють для вмісту всіх інших пакетів у цьому наборі. За умовчанням один пакет створюється у цьому наборі для кожного XSD у GML прикладній схемі, включаючи такі, що прямо чи непрямо імпортовані з XML просторів імен, відмінних від цільового простору імен для GML прикладної схеми, за виключенням XSD для GML просторів імен. Пакету вищого рівня прямо чи непрямо належать всі елементи UML моделі, перекладені з типів об'єктів у GML прикладній схемі.

Декларації GML прикладної схеми можна організувати в іншій структурі пакету, якщо пакет вищого рівня зберігає його ім'я і стереотип, а всі елементи моделі прямо чи непрямо належать до цього пакету.

Декларації типу та елементу в GML прикладній схемі переходять у визначення класів у UML прикладній схемі, тому структури елементів у GML XML документі можна перекласти в відповідні об'єкти в моделі екземпляру.

UML модель має містити в пакеті з іменем "ISO 19100" застосовні нормативні пакети комплексу міжнародних стандартів ISO 19100 або строгий профіль цієї моделі.

UML модель має містити UML пакет усіх інших GML прикладних схем, імпортованих GML прикладною схемою.

У таблиці E.1 наданий огляд, а повні деталі переходу визначені в подальших підрозділах.

**Таблиця E.1** – Огляд кодування схеми

<b>Огляд переходу із GML в UML прикладну схему</b>	
<b>GML прикладна схема</b>	<b>UML прикладна схема</b>
GML прикладна схема	Пакет <<ApplicationSchema>>
Документ {name} XSD GML схеми	Пакет, названий named {name}
Тип об'єкту та властивості та глобальний елемент для будь-якого типу об'єкту, що є прямим чи непрямым розширенням для gml:AbstractFeatureType, відмінний від тих, які розширюють gml:AbstractFeatureType	Клас із стереотипом <<FeatureType>>
Тип об'єкту та властивості та глобальний елемент для будь-якого типу об'єкту, що є прямим чи непрямым розширенням для gml:AbstractGMLType, відмінний від тих, які розширюють gml:AbstractFeatureType	Клас без стереотипу
Тип даних та властивості та глобальний	Клас із стереотипом <<DataType>>

Огляд переходу із GML в UML прикладну схему	
GML прикладна схема	UML прикладна схема
елемент для будь-якого типу об'єкту, що не є прямим чи непрямим розширенням для <code>gml:AbstractGMLType</code> , модель контенту якого є послідовністю властивостей	
Обмеження для <code>xsd:string</code> із значеннями списку	Клас із стереотипом <<Enumeration>>
Об'єднання переліку та шаблону	Клас із стереотипом <<CodeList>>
Тип даних та властивості та глобальний елемент для будь-якого типу об'єкту, що не є прямим чи непрямим розширенням <code>gml:AbstractGMLType</code> , модель контенту якого є вибором властивостей	Клас із стереотипом <<Union>>
Локальний <code>xsd:element</code> типів <code>simpleType</code> або <code>complexType</code> із простим контентом <code>simpleContent</code> або типом, що не є прямо чи непрямим успадкованим від <code>gml:AbstractGMLType</code>	UML атрибут
Локальний <code>xsd:element</code> типу, що містить <code>gml:AssociationAttributeGroup</code>	Роль в UML асоціації
Schematron обмеження	Не закодоване

### Е.2.3.2 Документи GML схеми

Пакет вищого рівня із стереотипом <<Application Schema>> створюють для вмісту всіх інших пакетів, генерованих для GML прикладної схеми:

- тегові значення “targetNamespace” та “xmlns” застосовують до пакету <<ApplicationSchema>> із відповідними значеннями для цільового простору імен GML прикладної схеми, наприклад, “<http://www.myorg.com/myns>” та “myns”.
- тегове значення “version” застосовують до пакету <<ApplicationSchema>> із значенням за умовчанням “1.0”; якщо атрибут версії елемента `xsd:schema` схеми вищого рівня для GML прикладної схеми існує та містить непорожнє значення, то його значення замінює тегове значення за умовчанням;
- тегове значення “xsdDocument” установлюють для відносного імені файлу документу XML схеми.

За умовчанням UML пакет генерують для кожного вхідного документу схеми в GML прикладній схемі, включаючи ті прямо чи непрямим імпортовані з XML просторів імен, що відмінні від цільового простору імен GML прикладної схеми, за виключенням документів XML схеми з GML просторів імен. Альтернативно, окремий документ XML схеми можна також розділити на кілька UML пакетів.

Пакети генерують в пакеті <<ApplicationSchema>> для GML прикладної схеми з іменами, що відповідають іменам вхідних документів схеми.

Твердження `xsd:include` та `xsd:import` у кожному документі вхідної схеми використовують для визначення та встановлення залежностей пакетів, генерованих у пакеті <<Application Schema>>.

### **E.2.3.3 GML типи об'єктів**

Кожен GML тип об'єкту треба перекладати в UML клас.

Якщо тип об'єкту прямо чи непрямо походить від `gml:AbstractFeatureType`, то стереотипом класу має бути `<<FeatureType>>`, і в іншому випадку не слід установлювати жодних стереотипів.

Ім'я класу має бути тим же, що і ім'я глобального елемента GML типу об'єкту.

Клас має бути абстрактним тоді і тільки тоді, якщо GML тип об'єкту абстрактний.

Якщо тип об'єкту походить від іншого GML типу об'єкту, то клас успадковується від відповідного надкласу. Якщо базовий тип визначений у GML прикладній схемі чи іншій імпортованій GML прикладній схемі, то надклас є класом, що відповідає цьому GML типу об'єктів. Якщо базовий тип визначений у GML просторі імен, то надклас визначений у таблиці Г.2. Якщо базовий тип перелічено в третьому стовпці таблиці, то надклас є класом у першому стовпці в тому ж самому рядку.

GML властивості GML типу об'єкту мають переходити в атрибути та ролі в асоціаціях, як описано в E.2.3.9. Треба надавати тегове значення "sequenceNumber" всім UML атрибутам та ролям в асоціації, створеним у цьому переході з цілочисловими значеннями у порядку зростання, що відображає порядок властивостей у послідовності типу об'єкту.

### **E.2.3.4 GML типи об'єктів (імпортовані з GML схеми)**

Комплексні типи з GML простору імен, подані списком у лівому стовпці таблиці Г.2, треба перекладати у наперед визначені UML класи, реалізовані профілем GML стандартів ISO в галузі географічної інформації, у другому стовпці таблиці.

### **E.2.3.5 Базові типи**

Прості типи з XML схеми та GML простору імен, показані в правому стовпці таблиці Г.2, треба перекладати у наперед визначені UML класи, реалізовані профілем GML стандартів ISO в галузі географічної інформації, у лівому стовпці таблиці.

### **E.2.3.6 GML типи даних**

Кожен GML тип даних треба перекладати в UML клас. Ім'я класу має бути тим же, що і ім'я комплексного типу без суфіксу `Type`.

Якщо GML тип даних походить від іншого GML типу даних (базовий тип), то клас успадковується від відповідного надкласу.

Якщо властивості GML типу даних вбудовані в елемент `xsd:sequence`, то стереотип класу має бути `<<DataType>>`; якщо вони вбудовані в елемент `xsd:choice`, то стереотип класу має бути встановлений як `<<Union>>`.

GML властивості GML типу об'єкту треба перекладати в атрибути та ролі в асоціації, як визначено в E.2.3.9. Треба надавати тегове значення "sequenceNumber" всім UML атрибутам та ролям в асоціації, створеним у цьому переході з цілочисловими значеннями у порядку зростання, що відображає порядок властивостей у послідовності типу об'єкту.

### **E.2.3.7 Переліки**

Простий тип, визначений у GML прикладній схемі як обмеження, накладене на `xsd:string` із значеннями переліку, треба перекладати в клас із стереотипом `<<Enumeration>>` у UML прикладній схемі.

Ім'я класу має бути іменем простого типу.

Кожний facet `xsd:enumeration` без анотації `xsd:appInfo` із успадкованим елементом `gml:codeListValue` треба перекладати в UML атрибут із значенням як ім'я атрибуту.

Кожний facet `xsd:enumeration` з анотацією `xsd:appInfo` з успадкованим елементом `gml:codeListValue` треба перекладати в початкове значення UML атрибуту з тим же іменем, що і значення елементу `gml:codeListValue`. Якщо такий UML атрибут є в класі, то facet треба ігнорувати.

#### **Е.2.3.8 Списки кодів**

Простий тип визначений у GML прикладній схемі як об'єднання обмеження `xsd:pattern` із значенням "other:\w{2,}" , а перелік треба перекладати в клас із стереотипом <<CodeList>> в UML прикладній схемі.

Іменем класу має бути ім'я простого типу.

Кожний facet `xsd:enumeration` переліку без анотації `xsd:appInfo` із успадкованим елементом `gml:codeListValue` треба перекладати в UML атрибут із значенням як іменем атрибуту.

Кожний facet `xsd:enumeration` переліку з анотацією `xsd:appInfo` із успадкованим елементом `gml:codeListValue` треба перекладати в початкове значення UML атрибуту з тим же іменем, що і значення елементу `gml:codeListValue`. Якщо не існує жодного UML атрибуту в класі, facet треба ігнорувати.

#### **Е.2.3.9 GML властивості**

Якщо тип елементу властивості є:

- простим типом властивості GML типу даних, властивість треба перекладати в UML атрибут із відповідним типом як тип даних;
- типом властивості GML типу об'єкту (включеним чи через посилання), модель контенту якого прямо чи непрямо походить від `gml:AbstractMemberType`, то властивість треба перекладати в роль у UML асоціації UML агрегації до класу, яким поданий цільовий GML тип об'єкту; якщо модель контенту елементу властивості містить атрибут "owns" ("володіє") із фіксованим значенням "true" (через Schematron обмеження), то треба змінити UML агрегацію на UML композицію;
- типом властивості GML типу об'єкту (включеним чи через посилання), то властивість треба перекладати в роль у UML асоціації до класу, яким поданий цільовий тип GML об'єкту; якщо тип властивості підтримує тільки через посилання, то цільовий тип GML об'єкту треба визначити з убудованої анотації `xsd:appInfo` з успадкованим елементом `gml:targetElement`, що визначає кваліфіковане ім'я елементу цільового типу; тежове значення "inlineOrByReference" треба встановити як "inline" ("включено") для подань, що дозволяють тільки включене кодування значення властивості, та "byReference" ("через посилання") для подань, що дозволяють кодування значення властивості тільки шляхом посилань;
- типом властивості GML типу об'єкту (включеним чи через посилання), модель контенту якого прямо чи непрямо походить від `gml:AbstractMetadataPropertyType`, то UML атрибут чи роль в асоціації має містити тежове значення "isMetadata" із значенням "true" ("істина").

Ім'я UML атрибуту або ролі в асоціації має бути іменем GML елементу властивості.

Чисельність UML атрибутів та ролей в асоціації має походити від значень `minOccurs` та `maxOccurs` GML властивості.

Якщо елемент властивості має анотацію `xsd:appInfo` із успадкованим елементом `gml:reversePropertyName`, то роль в асоціації треба визначати як частину асоціації між двома класами, де інша роль в асоціації має ім'я, що дорівнює значенню елементу `gml:reversePropertyName`.

### Е.2.3.10 Документація

Елементи `xsd:annotation/xsd:documentation` XML схеми в GML прикладних схемах треба перекладати у тегові значення “documentation” у UML прикладній схемі.

## ДОДАТОК Ж (довідковий)

### ІНСТРУКЦІЇ З ПОДІЛУ GML СХЕМИ НА ПІДНАБОРИ

#### Ж.1 Загальні відомості

Для використання у поділі GML схеми на піднабори рекомендовано автоматизований підхід. Цей додаток містить довідкову XSLT референцну реалізацію інструменту піднабору GML схеми. Інструмент складається з трьох XSLT аркушів стилю; три аркуші стилю показані в підрозділах Ж.2, Ж.3 та Ж.4.

Для створення GML схеми піднабору з використанням цього інструменту потрібно:

- Перетворити `gml.xsd` із використанням `depends.xslt` та XSLT процесору для того, щоб генерувати `gml.dep`.

*Приклад 1.* Із використанням Xalan команда могла б бути

```
$ java org.apache.xalan.xslt.Process -IN ../base/gml.xsd -XSL depends.xslt -OUT gml.dep.
```

- Якщо XSLT processor, який використовують, не може пройти параметри для аркушу стилю, який є предметом обробки, відредагувати `gmlSubset.xslt` та змінити “wanted” (“бажаний”) параметр так, щоб він містив список, розділений комами (із задньою комою), глобальних типів та елементів, кваліфікованих за простором імен та потрібних у GML схемі під набору.

*Приклад 2.* Треба змінити

```
<xsl:param name="wanted">,</xsl:param>
```

на

```
<xsl:param name="wanted">gml:featureProperty,gml:lineStringProperty,gml:polygonProperty,</xsl:param>.
```

- перетворити `gml.dep` із використанням `gmlSubset.xslt`, параметр, що називають “wanted”, установити як список, розділений комами (із задньою комою), глобальних типів та елементів, кваліфікованих за простором імен та потрібних у GML схемі піднабору, та XSLT процесору для того, щоб генерувати `gmlSubset.xsd`, що буде містити глобальні типи та елементи, визначені в параметрі “wanted”, та всі глобальні типи та елементи, на яких вони прямо чи непрямо використовують;

- генерований `gmlSubset.xsd` включити імпорти для просторів імен, які називають “xlink”, якщо список “wanted”, включений або залежний від будь-якого атрибуту з відповідного простору імен. В іншому випадку, це є окрема GML схема піднабору, яка відповідає вимогам для GML профілів.

#### Ж.2 depends.xslt

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

```

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink">
<!-- =====
This stylesheet is designed to be used on gml.xsd to produce gml.dep
for use by the gml schema subset utility gmlSubset.xslt to produce a specialized
gmlSubset.xsd that contains only the specified types and elements, and the types
and elements on which they depend.
(Цей аркуш стилю розроблений для використання на gml.xsd, щоб генерувати
gml.dep для використання утилітою gml схеми піднабору gmlSubset.xslt, щоб
генерувати спеціалізований gmlSubset.xsd, що містить тільки визначені типи
та елементи, і типи та елементи, від яких вони залежать).
=====-->
<xsl:output method="xml" encoding="UTF-8" indent="yes"/>
<xsl:include href="utility.xslt"/>
<!-- NEWLINE = &#xA; -->
<xsl:param
name="schemas">gml.xsd,observation.xsd,dynamicFeature.xsd,coverage.xsd,topology.xsd,defaultStyle
e.xsd,coordinateReferenceSystems.xsd,feature.xsd,valueObjects.xsd,grids.xsd,geometryComplexes.xsd
,datums.xsd,coordinateSystems.xsd,coordinateOperations.xsd,geometryAggregates.xsd,referenceSyste
ms.xsd,dataQuality.xsd,geometryPrimitives.xsd,geometryBasic2d.xsd,direction.xsd,geometryBasic0d1
d.xsd,measures.xsd,temporal.xsd,units.xsd,dictionary.xsd,gmlBase.xsd,basicTypes.xsd,</xsl:param>
<xsl:param name="allSchemas">
<xsl:call-template name="getUniqueSchemaList">
<xsl:with-param name="list" select="$schemas"/>
<xsl:with-param name="usePre"></xsl:with-param>
</xsl:call-template>
</xsl:param>
<xsl:template match="/">
<xsl:param name="docName">gml.xsd</xsl:param>
<xsl:param name="top" select="true()"/>
<xsl:param name="tns" select="//xsd:schema/@targetNamespace"/>
<xsl:param name="vers" select="//xsd:schema/@version"/>
<xsl:variable name="ltns">
<xsl:for-each select="//xsd:schema/namespace::*">
<xsl:if test="local-name() != 'targetNamespace' and string() = $tns">
<xsl:value-of select="local-name()"/>
</xsl:if>
</xsl:for-each>
</xsl:variable>
<xsl:variable name="tnsp">
<xsl:choose>
<xsl:when test="$ltns = "">
<xsl:call-template name="getTargetNameSpacePrefix">
<xsl:with-param name="list" select="$tns"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$ltns"/>

```



```

    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<xsl:text>&#xA;</xsl:text>
<xsl:choose>
  <xsl:when test="$top">
    <xsl:text disable-output-escaping="yes">&lt;depends version="</xsl:text><xsl:value-of
select="$vers"/><xsl:text disable-output-escaping="yes">"&gt;</xsl:text>
  </xsl:when>
  <xsl:otherwise>
    <xsl:for-each select="/xsd:schema">
      <xsl:for-each select="xsd:complexType | xsd:group | xsd:simpleType | xsd:element |
xsd:attribute |xsd:attributeGroup">
        <xsl:variable name="type" select="local-name()"/>
        <xsl:choose>
          <xsl:when test="$type = 'complexType' ">
            <xsl:call-template name="complexType">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="$type = 'group' ">
            <xsl:call-template name="complexType">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="$type = 'simpleType' ">
            <xsl:call-template name="simpleType">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="$type = 'element' ">
            <xsl:call-template name="globalElement">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="$type = 'attribute' ">
            <xsl:call-template name="globalAtt">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="$type = 'attributeGroup' ">
            <xsl:call-template name="globalAtt">
              <xsl:with-param name="docName" select="$docName"/>
              <xsl:with-param name="targetNamespace" select="$tnsp"/>
            </xsl:call-template>
          </xsl:when>
        </xsl:choose>
      </xsl:for-each>
    </xsl:for-each>
  </xsl:otherwise>
</xsl:choose>

```

```

        </xsl:call-template>
        </xsl:when>
        <xsl:otherwise/>
    </xsl:choose>
</xsl:for-each>
</xsl:for-each>
</xsl:otherwise>
</xsl:choose>
<xsl:if test="$stop">
    <xsl:call-template name="dependSchemas">
        <xsl:with-param name="list" select="$allSchemas"/>
    </xsl:call-template>
    <xsl:text disable-output-escaping="yes">&#xA;&lt;/depends&gt;&#xA;</xsl:text>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="complexType">
    <xsl:param name="docName"/>
    <xsl:param name="targetNamespace"/>
    <xsl:variable name="name" select="@name"/>
    <xsl:if test="$name">
        <xsl:element name="def">
            <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/><xsl:value-of
select="$name"/></xsl:attribute>
            <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
            <xsl:variable name="uses">
                <xsl:apply-templates select="./xsd:complexContent|./xsd:simpleContent"/>
                <xsl:call-template name="EltAndAtt"/>
            </xsl:variable>
            <!-- USES <xsl:value-of select="$uses"/> -->
            <xsl:call-template name="writeUses">
                <xsl:with-param name="list" select="$uses"/>
            </xsl:call-template>
        </xsl:element>
    </xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template match="xsd:complexContent">
    <xsl:for-each select="descendant::xsd:extension">
        <xsl:value-of select="@base"/>
        <xsl:text>?extension|</xsl:text>
    </xsl:for-each>
    <xsl:for-each select="descendant::xsd:restriction">
        <xsl:value-of select="@base"/>
        <xsl:text>?restriction|</xsl:text>
    </xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template match="xsd:simpleContent">

```

```

<xsl:for-each select="descendant::xsd:extension">
  <xsl:value-of select="@base"/>
  <xsl:text>?extension|</xsl:text>
</xsl:for-each>
<xsl:for-each select="descendant::xsd:restriction">
  <xsl:value-of select="@base"/>
  <xsl:text>?restriction|</xsl:text>
</xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template name="EltAndAtt">
  <xsl:for-each select="descendant::xsd:element | descendant::xsd:group | descendant::xsd:attribute
|
descendant::xsd:attributeGroup">
  <xsl:variable name="name" select="@type | @ref"/>
  <xsl:if test="$name and contains($name,':')">
    <xsl:value-of select="$name"/>
    <xsl:text>|</xsl:text>
  </xsl:if>
</xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template name="simpleType">
  <xsl:param name="docName"/>
  <xsl:param name="targetNamespace"/>
  <xsl:variable name="name" select="@name"/>
  <xsl:if test="$name">
    <xsl:element name="def">
      <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
      <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
      <!-- SIMPLE <xsl:copy-of select="."/>-->
      <xsl:variable name="uses">
        <xsl:for-each select="xsd:union">
          <!-- UNION <xsl:value-of select="@memberTypes"/> -->
          <xsl:variable name="members" select="@memberTypes"/>
          <xsl:if test="$members">
            <xsl:value-of select="translate($members,',|')"/>
            <xsl:text>|</xsl:text>
          </xsl:if>
        </xsl:for-each>
        <xsl:for-each select="xsd:list">
          <xsl:variable name="items" select="@itemType"/>
          <xsl:if test="$items">
            <xsl:value-of select="$items"/>
            <xsl:text>|</xsl:text>
          </xsl:if>
        </xsl:for-each>
      </xsl:variable>

```

```

    <!-- USES <xsl:value-of select="$uses"/> -->
    <xsl:call-template name="writeUses">
      <xsl:with-param name="list" select="$uses"/>
    </xsl:call-template>
  </xsl:element>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="globalElement">
  <xsl:param name="docName"/>
  <xsl:param name="targetNamespace"/>
  <xsl:variable name="name" select="@name"/>
  <xsl:if test="$name">
    <xsl:element name="def">
      <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
      <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
      <xsl:variable name="uses">
        <xsl:variable name="type" select="@type"/>
        <xsl:if test="$type and contains($type,':')">
          <xsl:value-of select="$type"/>
          <xsl:text>|</xsl:text>
        </xsl:if>
        <xsl:variable name="sub" select="@substitutionGroup"/>
        <xsl:if test="$sub">
          <xsl:value-of select="$sub"/>
          <xsl:text>|</xsl:text>
        </xsl:if>
      </xsl:variable>
      <!-- USES <xsl:value-of select="$uses"/> -->
      <xsl:call-template name="writeUses">
        <xsl:with-param name="list" select="$uses"/>
      </xsl:call-template>
    </xsl:element>
  </xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="globalAtt">
  <xsl:param name="docName"/>
  <xsl:param name="targetNamespace"/>
  <xsl:variable name="name" select="@name"/>
  <xsl:if test="$name">
    <xsl:element name="def">
      <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
      <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
      <xsl:variable name="uses">
        <xsl:variable name="type" select="@type"/>
        <xsl:if test="$type and contains($type,':')">

```

```

        <xsl:value-of select="$type"/>
        <xsl:text>|</xsl:text>
    </xsl:if>
    <xsl:call-template name="EltAndAtt"/>
</xsl:variable>
<!-- USES <xsl:value-of select="$uses"/> -->
<xsl:call-template name="writeUses">
    <xsl:with-param name="list" select="$uses"/>
</xsl:call-template>
</xsl:element>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="writeUses">
    <xsl:param name="list"/>
    <xsl:if test="$list != "">
        <xsl:variable name="first" select="substring-before($list, '|")"/>
        <xsl:variable name="eor" select="substring-after($first, '?")"/>
        <xsl:variable name="use">
            <xsl:choose>
                <xsl:when test="contains($first, '?")">
                    <xsl:value-of select="substring-before($first, '?")"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="$first"/>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:variable>
        <xsl:variable name="testp">
            <xsl:value-of select="$use"/>
            <xsl:text>|</xsl:text>
        </xsl:variable>
        <xsl:variable name="testq">
            <xsl:value-of select="$use"/>
            <xsl:text>?</xsl:text>
        </xsl:variable>
        <xsl:variable name="rest" select="substring-after($list, '|")"/>
        <xsl:choose>
            <xsl:when test="contains($rest, $testp)"/>
            <xsl:when test="contains($rest, $testq)"/>
            <xsl:when test="$use = """/>
            <xsl:otherwise>
                <xsl:element name="uses">
                    <xsl:attribute name="name"><xsl:value-of select="$use"/></xsl:attribute>
                    <xsl:if test="$eor != "">
                        <xsl:attribute name="derivation"><xsl:value-of select="$eor"/> </xsl:attribute>
                    </xsl:if>
                </xsl:element>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:if>
</xsl:template>

```

```

    </xsl:choose>
    <xsl:call-template name="writeUses">
      <xsl:with-param name="list" select="$rest"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="dependSchemas">
  <xsl:param name="list"/>
  <xsl:if test="$list != "">
    <xsl:variable name="first" select="substring-before($list, ',')"/>
    <xsl:variable name="rest" select="substring-after($list, ',')"/>
    <xsl:apply-templates select="document($first, /)">
      <xsl:with-param name="docName" select="$first"/>
      <xsl:with-param name="top" select="false()"/>
    </xsl:apply-templates>
    <xsl:choose>
      <xsl:when test="contains($rest, ',')">
        <xsl:call-template name="dependSchemas">
          <xsl:with-param name="list" select="$rest"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
  </xsl:if>
</xsl:template>
<!-- ===== -->
<!-- ===== -->
</xsl:stylesheet>

```

### Ж.3 gmlSubset.xslt

```

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
  <!-- =====
  This stylesheet is designed to be used on gml.dep (produced from
  gml.xsd by depends.xslt) to produce a specialized gmlSubset.xsd that
  contains only the types and elements specified in the "wanted" parameter,
  and the types and elements on which they depend. Note that the type and
  element items in the "wanted" parameter must include namespace prefixes,
  and that they must be separated by commas, including a trailing comma after
  the last item.
  Аркуш стилів розроблений для використання на gml.dep (генерований від
  gml.xsd із допомогою depends.xslt) для генерування спеціалізованого
  gmlSubset.xsd, що містить тільки типи та елементи, визначені в параметрі
  "wanted", та тип і елементи, які залежать від них. Треба відзначити, що предмети

```

піднабору, що є типами та елементами у параметрі "wanted" мають включати префікси просторів імен, та їх треба відокремлювати комами, включно із *задньою* комою після останнього предмету.

```

=====-->
<xsl:include href="utility.xslt"/>
<xsl:param name="baseUri" select="document('../base/gml.xsd')"/>
<!-- sample1 <xsl:param name="wanted">gml:featureProperty,gml:lineStringProperty,gml:polygonProperty,</xsl:param> -->
<!-- sample2 <xsl:param name="wanted">gml:GeodeticCRS,gml:AbstractCoverage,gml:track,</xsl:param> -->
<!-- sample3 <xsl:param
name="wanted">gml:AbstractFeatureCollection,gml:ItemStyleDescriptorType,gml:FeatureConstraintType,</xsl:param> -->
<xsl:param
name="wanted">gml:metaDataProperty,gml:Abstractassociation,gml:members,gml:Array,gml:curvePr
operty,gml:LineString,g
ml:LinearRing,gml:exterior,gml:interior,gml:surfaceMember,gml:surfaceProperty,gml:multiSurfacePro
perty,gml:directedNode,gml:directedEdge,gml:directedFace,gml:IsolatedProperty,gml:featureProperty,
gml:featureMembers,gml:AbstractFeatureCollection,gml:featureMember,gml:BaseStyleDescriptorTyp
e,</xsl:param>
<xsl:template match="/">
  <xsl:variable name="wantedList">
    <xsl:call-template name="getWantedList">
      <xsl:with-param name="list" select="$wanted"/>
      <xsl:with-param name="from">BEGIN</xsl:with-param>
      <xsl:with-param name="depth">0</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="vers" select="//depends/@version"/>
  <schema targetNamespace="http://www.opengis.net/gml/3.2"
xmlns="http://www.w3.org/2001/XMLSchema"
"http://purl.oclc.org/dsdl/schematron" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="{ $vers } ">
    <annotation>
      <documentation>GML Subset schema for <xsl:value-of select="$wanted"/> written by
gmlSubset.xslt.
</documentation>
    </annotation>

    <xsl:if test="contains($wantedList,'xlink:')">
      <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="../xlink/xlinks.xsd"/>
    </xsl:if>
    <xsl:call-template name="writeWantedList">
      <xsl:with-param name="list" select="$wantedList"/>
    </xsl:call-template>
  </schema>
</xsl:template>
<!-- ===== -->
<xsl:template name="getDocName">
  <xsl:param name="wanted"/>
  <xsl:for-each select="//depends/def[@name=$wanted]">

```

```

        <xsl:value-of select="@doc"/>
    </xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template name="getUses">
    <xsl:param name="wanted"/>
    <xsl:for-each select="//depends/def[@name=$wanted]">
        <xsl:for-each select="uses">
            <xsl:value-of select="@name"/>
            <xsl:text>,</xsl:text>
        </xsl:for-each>
    </xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template name="writeWanted">
    <xsl:param name="wanted"/>
    <xsl:choose>
        <xsl:when test="contains($wanted,'xlink:')">
            <!-- XLINK <xsl:value-of select="$wanted"/> -->
        </xsl:when>
        <xsl:otherwise>
            <!-- OTHER <xsl:value-of select="$wanted"/> -->
            <xsl:variable name="docName">
                <xsl:call-template name="getDocName">
                    <xsl:with-param name="wanted" select="$wanted"/>
                </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="localName">
                <xsl:call-template name="removePrefix">
                    <xsl:with-param name="name" select="$wanted"/>
                    <xsl:with-param name="pre">:</xsl:with-param>
                </xsl:call-template>
            </xsl:variable>
            <xsl:call-template name="Separator"/>
            <xsl:for-each select="document($docName,$baseUri)">
                <xsl:for-each select="//xsd:schema/*[@name = $localName]">
                    <xsl:copy-of select="."/>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>
<!-- ===== -->
<xsl:template name="writeWantedList">
    <xsl:param name="list"/>
    <xsl:if test="$list != "">
        <xsl:variable name="first" select="substring-before($list, ',')"/>
        <xsl:variable name="rest" select="substring-after($list, ',')"/>
        <xsl:call-template name="writeWanted">

```



```

        <xsl:with-param name="wanted" select="$first"/>
    </xsl:call-template>
    <xsl:if test="contains($rest,',')">
        <xsl:call-template name="writeWantedList">
            <xsl:with-param name="list" select="$rest"/>
        </xsl:call-template>
    </xsl:if>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="getWantedList">
    <xsl:param name="list"/>
    <xsl:param name="seen"/>
    <xsl:param name="from"/>
    <xsl:param name="depth"/>
    <xsl:if test="$list != "">
        <xsl:variable name="first" select="substring-before($list, '/')"/>
        <xsl:variable name="firstSep" select="concat($first, '/')"/>
        <xsl:variable name="rest" select="substring-after($list, '/')"/>
        <xsl:choose>
            <xsl:when test="contains($seen,$firstSep)">
                <xsl:call-template name="getWantedList">
                    <xsl:with-param name="list" select="$rest"/>
                    <xsl:with-param name="seen" select="$seen"/>
                    <xsl:with-param name="from">REST</xsl:with-param>
                    <xsl:with-param name="depth" select="$depth + 1"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$firstSep"/>
                <xsl:variable name="usesList">
                    <xsl:call-template name="getUses">
                        <xsl:with-param name="wanted" select="$first"/>
                    </xsl:call-template>
                </xsl:variable>
                <xsl:variable name="toDo" select="concat($usesList,$rest)"/>
                <xsl:variable name="nowSeen" select="concat($seen,$firstSep)"/>
                <xsl:call-template name="getWantedList">
                    <xsl:with-param name="list" select="$toDo"/>
                    <xsl:with-param name="seen" select="$nowSeen"/>
                    <xsl:with-param name="from">USES</xsl:with-param>
                    <xsl:with-param name="depth" select="$depth + 1"/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:if>
</xsl:template>
<!-- ===== -->
</xsl:stylesheet>

```

**Ж.4 utility.xslt**

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
  <!-- ===== -->
  <xsl:template name="getTargetNameSpacePrefix">
    <xsl:param name="list"/>
    <xsl:if test="$list != "">
      <xsl:variable name="first" select="substring-before($list, '/')"/>
      <xsl:variable name="rest" select="substring-after($list, '/')"/>
      <xsl:choose>
        <xsl:when test="contains($rest, '/')">
          <xsl:call-template name="getTargetNameSpacePrefix">
            <xsl:with-param name="list" select="$rest"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:when test="$rest = "">
          <xsl:value-of select="$first"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="$rest"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:if>
  </xsl:template>
  <!-- ===== -->
  <xsl:template name="getPathPrefix">
    <xsl:param name="file"/>
    <xsl:if test="contains($file, '/')">
      <xsl:variable name="pre" select="substring-before($file, '/')"/>
      <xsl:variable name="suf" select="substring-after($file, '/')"/>
      <xsl:choose>
        <xsl:when test="contains($suf, '/')">
          <xsl:value-of select="$pre"/><xsl:text></xsl:text>
          <xsl:call-template name="getPathPrefix">
            <xsl:with-param name="file" select="$suf"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:variable name="path">
            <xsl:call-template name="removeSuffix">
              <xsl:with-param name="name" select="$file"/>
              <xsl:with-param name="suf" select="$suf"/>
            </xsl:call-template>
          </xsl:variable>
          <xsl:value-of select="$path"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:if>
  </xsl:template>

```

```

    </xsl:choose>
  </xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="removePrefix">
  <xsl:param name="name"/>
  <xsl:param name="pre"/>
  <xsl:variable name="npName">
    <xsl:choose>
      <xsl:when test="contains($name,$pre)">
        <xsl:value-of select="substring-after($name,$pre)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$name"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:value-of select="$npName"/>
</xsl:template>
<!-- ===== -->
<xsl:template name="removeSuffix">
  <xsl:param name="name"/>
  <xsl:param name="suf"/>
  <xsl:variable name="nsName">
    <xsl:choose>
      <xsl:when test="contains($name,$suf)">
        <xsl:value-of select="substring-before($name,$suf)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$name"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:value-of select="$nsName"/>
</xsl:template>
<!-- ===== -->
<xsl:template name="lowerLeading">
  <xsl:param name="name"/>
  <xsl:variable name="ch1" select="substring($name, 1, 1)"/>
  <xsl:variable name="lc1"
select="translate($ch1,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz)"/>
  <xsl:value-of select="concat($lc1, substring($name, 2))"/>
</xsl:template>
<!-- ===== -->
<xsl:template name="uniqueList">
  <xsl:param name="list"/>
  <xsl:param name="sep"/>
  <xsl:param name="seen"/>
  <xsl:param name="pre">../base/</xsl:param>

```

```

<xsl:if test="$list != "">
  <xsl:variable name="first" select="substring-before($list, $sep)"/>
  <xsl:variable name="firstSep" select="concat($first,$sep)"/>
  <xsl:variable name="rest" select="substring-after($list, $sep)"/>
  <xsl:choose>
    <xsl:when test="contains($seen,$firstSep)">
      <xsl:call-template name="uniqueList">
        <xsl:with-param name="list" select="$rest"/>
        <xsl:with-param name="sep" select="$sep"/>
        <xsl:with-param name="seen" select="$seen"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$firstSep"/>
      <xsl:variable name="nowSeen" select="concat($seen, $firstSep)"/>
      <xsl:call-template name="uniqueList">
        <xsl:with-param name="list" select="$rest"/>
        <xsl:with-param name="sep" select="$sep"/>
        <xsl:with-param name="seen" select="$nowSeen"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="getIncludedDocs">
  <xsl:param name="docName"/>
  <xsl:param name="usePre"/>
  <xsl:param name="seenList"/>
  <xsl:param name="sep"/>,</xsl:param>
  <xsl:value-of select="$docName"/>
  <xsl:text>,</xsl:text>
  <xsl:variable name="pathPre">
    <xsl:call-template name="getPathPrefix">
      <xsl:with-param name="file" select="$docName"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="callPathPre">
    <xsl:choose>
      <xsl:when test="$pathPre = " or $pathPre = './' ">
        <xsl:value-of select="$usePre"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$pathPre"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:for-each select="document($docName, /)">
    <xsl:for-each select="//xsd:include |//xsd:import">

```

```

<xsl:variable name="iDoc" select="@schemaLocation"/>
<xsl:variable name="iPathPre">
  <xsl:call-template name="getPathPrefix">
    <xsl:with-param name="file" select="$iDoc"/>
  </xsl:call-template>
</xsl:variable>
<xsl:variable name="iDocSuf">
  <xsl:call-template name="removePrefix">
    <xsl:with-param name="name" select="$iDoc"/>
    <xsl:with-param name="pre" select="$iPathPre"/>
  </xsl:call-template>
</xsl:variable>
<xsl:variable name="usePathPre">
  <xsl:choose>
    <xsl:when test="$iPathPre = " or $iPathPre = './' ">
      <xsl:value-of select="$callPathPre"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$iPathPre"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<xsl:variable name="uDoc">
  <xsl:value-of select="concat($usePathPre,$iDocSuf)"/>
</xsl:variable>
<xsl:variable name="uDocSep">
  <xsl:value-of select="concat($uDoc,$sep)"/>
</xsl:variable>
<xsl:if test="not(contains($seenList,$uDocSep))">
  <xsl:variable name="seenListPlus" select="concat($seenList,$uDocSep)"/>
  <xsl:call-template name="getIncludedDocs">
    <xsl:with-param name="docName" select="$uDoc"/>
    <xsl:with-param name="usePre" select="$usePathPre"/>
    <xsl:with-param name="seenList" select="$seenListPlus"/>
  </xsl:call-template>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
</xsl:template>
<!-- ===== -->
<xsl:template name="getDocumentList">
  <xsl:param name="list"/>
  <xsl:param name="seenList"/>
  <xsl:param name="usePre"/>
  <xsl:if test="$list != "">
    <xsl:variable name="first" select="substring-before($list, ',')"/>
    <xsl:variable name="rest" select="substring-after($list, ',')"/>
    <xsl:variable name="included">
      <xsl:call-template name="getIncludedDocs">

```

```

        <xsl:with-param name="docName" select="$first"/>
        <xsl:with-param name="usePre" select="$usePre"/>
        <xsl:with-param name="seenList" select="$seenList"/>
    </xsl:call-template>
</xsl:variable>
<xsl:value-of select="$included"/>
<xsl:variable name="seenListIncluded" select="concat($seenList,$included)"/>
<xsl:if test="contains($rest,',')">
    <xsl:call-template name="getDocumentList">
        <xsl:with-param name="list" select="$rest"/>
        <xsl:with-param name="seenList" select="$seenListIncluded"/>
        <xsl:with-param name="usePre" select="$usePre"/>
    </xsl:call-template>
</xsl:if>
</xsl:if>
</xsl:template>
<!-- ===== -->
<xsl:template name="getUniqueSchemaList">
    <xsl:param name="list"/>
    <xsl:param name="usePre"/>
    <xsl:variable name="allSchemas">
        <xsl:call-template name="getDocumentList">
            <xsl:with-param name="list" select="$list"/>
            <xsl:with-param name="usePre" select="$usePre"/>
        </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="uniqueSchemas">
        <xsl:call-template name="uniqueList">
            <xsl:with-param name="list" select="$allSchemas"/>
            <xsl:with-param name="sep">,</xsl:with-param>
        </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="$uniqueSchemas"/>
</xsl:template>
<!-- ===== -->
    <xsl:template name="Separator">
    <xsl:param name="comment" select="''"/>
    <xsl:text>&#xA;</xsl:text>
    <xsl:text disable-output-escaping="yes">
        &lt;!-- =====</xsl:text><xsl:value-of select="$comment"/><xsl:text disable-
outputescaping="
yes">===== --&gt;
    </xsl:text>
    <!-- ===== -->
</xsl:template>
</xsl:stylesheet>

```

ДОДАТОК И  
(довідковий)  
**СТИЛІЗАЦІЯ ЗА УМОВЧАННЯМ**

### **И.1 Загальні відомості**

GML було розроблено для чіткого відокремлення контенту даних від графічного чи іншого подання цих даних. Тому описи GML географічного об'єкту не містять будь-якої інформації відносно подання цього об'єкту.

У цьому додатку надано компоненти схеми для визначення наборів правил стилізації, які при застосуванні до асоційованого GML набору даних генерують графічну візуалізацію цих даних із використанням W3C SVG. Ці правила стилізації уможливають створення SVG документів на основі елементів даних включно з іменами типу географічного об'єкту, тематичних та просторових властивостей об'єкту і т. і.

Можливість, що дозволяє визначати стилі для GML даних, розглядають як суттєву для зображення GML даних. Компоненти схеми стилізації за умовчанням забезпечують засоби для цього, однак, є відомі питання, і особливо помітно, що потрібна гармонізація з існуючим і ширше реалізованим дескриптором стилів шарів (SLD) специфікації реалізації OpenGIS. Потенційно треба враховувати в цьому кроці також перегляд ISO 19117. Крім того, треба враховувати в цьому процесі додаткові вдосконалення, а саме впровадження словника стилів для яснішого відокремлення географічних об'єктів від їх інформації про стилізацію. Тому компоненти схеми стилів за умовчанням не є нормативними в цьому міжнародному стандарті.

Треба зауважити, що не вважають за важливе, щоб ці компоненти схеми опису стилю знаходилися в GML просторі імен, однак, важливо, щоб існувала можливість стандартизованого опису стилю. Коли з'явиться загальноприйнята схема, цей додаток можна усунути.

Компоненти схеми стилю, описані в цьому додатку, призначені для використання як окрема модель, яку можна підключити до GML набору даних.

*Приклад 1.* Типовий підхід забезпечив би стабільний стиль, асоційований із конкретним типом географічного об'єкту.

Термін “за умовчанням” означає широке відношення до асоційованих GML даних, та інформацію про стиль, призначений цьому набору даних, можна використовувати для стилізації, але можна повністю ігнорувати. Використання правил асоційованої стилізації за умовчанням таким чином визначено аплікацією стилізації.

Стиль, як він визначений у цьому додатку, розуміють як асоціацію між GML об'єктом (тобто, географічним об'єктом, геометрією чи топологією) та елементом графічного подання, вираженого у SVG. Наприклад, стиль може виражати це графічне подання для `gml:Curve`, що репрезентує осьову лінію дорожнього об'єкту та має бути в SVG шляху з особливими шириною та кольором для SVG шляху.

Компоненти схеми стилю за умовчанням також залежать від схем синхронізованої мови інтеграції мультимедій W3C (Synchronized Multimedia Integration Language).

Відношення інформації про стиль за умовчанням та екземплярів GML даних досягають через властивість `gml:defaultStyle`. Цю властивість можна призначати екземпляру шляхом її призначення типу географічного об'єкту в асоційований

прикладній схемі. Оскільки GML є кодуванням, базованим на географічних об'єктах, стиль за умовчанням завжди застосовують до географічного об'єкту, об'єктів чи їх колекцій. Стилізація за умовчанням уможливорює таке графічне подання цих географічних об'єктів, базоване на їх властивостях.

**Примітка 1.** Концептуальна модель для схеми стилізації за умовчанням та опис реалізації ISO 19117 не входить у цей додаток, оскільки він довідковий у цьому міжнародному стандарті.

## И.2 Елементи стилізації вищого рівня

### И.2.1 Огляд

Зв'язок між GML набором даних та описом стилізації встановлюють через одну властивість, `gml:defaultStyle`. Значення цієї властивості, об'єкт `gml:Style`, містить всі описи стилізації. Властивість `gml:defaultStyle` треба визначати у визначенні типу географічного об'єкту в прикладній схемі так, щоб можна було її використовувати для асоціювання елемента географічного об'єкту з правилами стилізації.

### И.2.2 `defaultStyle`

Властивість `gml:defaultStyle` – це властивість, визначена як глобальний елемент, яку можна призначити будь-якому географічному об'єкту, визначеному в прикладній схемі. Визначення властивості є таким:

```
<element name="defaultStyle" type="gml:DefaultStylePropertyType"/>

<complexType name="DefaultStylePropertyType">
  <sequence minOccurs="0">
<element ref="gml:AbstractStyle"/>
  </sequence>
  <attribute name="about" type="anyURI"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Властивість `gml:defaultStyle` може містити атрибут `about`. Це можна використовувати у колекції географічних об'єктів для призначення стилів за умовчанням об'єктам у цій колекції. У такому випадку властивість `gml:defaultStyle` приєднують до колекції (прикладна схема може допускати будь-яке число таких властивостей), причому кожний включає або посилається на інформацію про правило стилізації. Якщо використовують атрибут `about`, то він може посилатися на будь-який географічний об'єкт (чи будь-яку колекцію географічних об'єктів); якщо його не використовують, то стиль географічного об'єкту застосовують до батьківського географічного об'єкту властивості `gml:defaultStyle`, до якої приєднано атрибут `about`.

Цю властивість можна включати в географічний об'єкт через прикладну схему, яка визначає тип об'єкту.

*Приклад 1.* У наступному визначенні типу географічного об'єкту “дорога” (“road”) `exp:Road` показано включення властивості `gml:defaultStyle`:

```
<element name="Road" type="exp:RoadType" substitutionGroup="gml:AbstractFeature"/>

<complexType name="RoadType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
```



```

<sequence>
  <element ref="gml:centerLineOf"/>
  <element ref="gml:defaultStyle"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

### И.2.3 Стиль

Об'єкт `gml:Style` є конкретним значенням за умовчанням властивості `gml:defaultStyle`. Це є об'єктом стилізації вищого рівня, що інкапсулює всі інші, часткові описи стилю. Це визначення є таким:

```

<element name="Style" type="gml:StyleType" substitutionGroup="gml:AbstractStyle"/>

<complexType name="StyleType">
  <complexContent>
    <extension base="gml:AbstractStyleType">
      <sequence>
        <element ref="gml:featureStyle" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:graphStyle" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Модель контенту об'єкту `gml:Style` отримують шляхом розширення від `gml:AbstractStyleType`. Цей тип виступає абстрактним базовим типом для цілей розширюваності, тобто створення користувачем об'єктів стилю, та не додає жодного нового контенту до `gml:AbstractGMLType`, від якого він походить:

```

<element name="AbstractStyle" type="gml:AbstractStyleType" abstract="true"
  substitutionGroup="gml:AbstractGML"/>

<complexType name="AbstractStyleType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType"/>
  </complexContent>
</complexType>

```

Однак, немає припущення, що створення об'єктів стилю користувачем буде поширеною практикою, оскільки `gml:Style` забезпечує багаті можливості для опису стилів.

Визначення об'єкту `gml:Style`, подане раніше в тексті, є самим по собі належним прикладом використання механізму розширюваності та показує, як можна застосовувати в схемі ці правила:

- модель контенту конкретного об'єкту стилю походить від `gml:AbstractStyleType`;
- конкретний об'єкт стилю може замінювати `gml:AbstractStyle`.

Функція елементів стилізації об'єкту `gml:Style`, а саме `gml:featureStyle`, `gml:graphStyle` призначено для опису стилів для двох аспектів GML даних:

індивідуальних географічних об'єктів та графів топології, що складаються з колекцій географічних об'єктів. Треба зауважити, що елементи, які описують стилі для певних аспектів географічних об'єктів, а саме, стиль об'єкту, стиль графу, стиль геометрії, стиль топології та стиль написів, часто називають дескрипторами стилів.

### И.3 Стиль географічного об'єкту

#### И.3.1 FeatureStyle

Дескриптор стилю географічного об'єкту призначають для `gml:Style` через властивість `gml:featureStyle`. Вона дозволяє, подібно до інших GML властивостей, визначати значення шляхом включення чи віддалено:

```
<element name="featureStyle" type="gml:FeatureStylePropertyType"/>

<complexType name="FeatureStylePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:FeatureStyle"/>
  </sequence>
  <attribute name="about" type="anyURI"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Її значенням є `gml:FeatureStyle` – дескриптор стилю географічного об'єкту. Дескриптор стилю географічного об'єкту описує інформацію про стилізацію для набору екземплярів географічних об'єктів. Набір визначають із допомогою механізмів вибірки, що входять до цього дескриптору. Стиль застосовують до кожного географічного об'єкту в наборі незалежно – жодні відношення, що можуть існувати між об'єктами в наборі, не є значущими.

**Примітка 1.** Протилежним випадком є такий, за якого стиль застосовують до цілого набору об'єктів.

Визначення дескриптору стилю географічного об'єкту є таким:

```
<element name="FeatureStyle" type="gml:FeatureStyleType" substitutionGroup="gml:AbstractGML"/>

<complexType name="FeatureStyleType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element name="featureConstraint" type="string" minOccurs="0"/>
        <element ref="gml:geometryStyle" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:topologyStyle" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:labelStyle" minOccurs="0"/>
      </sequence>
      <attribute name="featureType" type="string"/>
      <attribute name="baseType" type="string"/>
      <attribute name="queryGrammar" type="gml:QueryGrammarEnumeration"/>
    </extension>
  </complexContent>
</complexType>
```

Екземпляри географічних об'єктів, до яких застосований стиль, вибирають із використанням одного з атрибутів `featureType` або `baseType` та елементу `gml:featureConstraint`. Ці два атрибути треба використовувати виключно, з елементом `gml:featureConstraint` чи без нього.

**Примітка 2.** У перегляді елементи властивості `gml:featureConstraint` мають стати атрибутом.

### И.3.2 `featureType`

Найпростішим та найпоширенішим способом зв'язування географічних об'єктів та стилів є використання цього атрибуту. Його значення буде задекларованим іменем географічного об'єкту, екземпляри якого треба стилізувати.

*Приклад 1.* Якщо значення становить `exp:Road`, то об'єкт `gml:FeatureStyle` буде просто застосований до всіх об'єктів доріг (`Road`). Значення цього атрибуту є завжди іменем елементу з прикладної схеми, яка декларує даний географічний об'єкт.

### И.3.3 `baseType`

Інший шлях вибору екземплярів географічного об'єкту, до яких застосований стиль, полягає в визначенні (як значення цього атрибуту) імені базового типу, з якого походять дані географічні об'єкти. Це завжди є іменем комплексного типу XML схеми. Можна використовувати будь-який комплексний тип із ланцюгу походження; стиль застосовують до будь-якого екземпляру географічного об'єкту, від якого він нарешті походить.

*Приклад 1.* Якщо `gml:AbstractFeatureType` використовують як значення атрибуту, стиль застосовують до всіх екземплярів географічного об'єкту в наборі даних.

### И.3.4 `featureConstraint`

Цю властивість використовують для подальшого обмеження набору екземплярів географічного об'єкту, до яких застосовувати стиль. Вона не є обов'язковою, а її значенням є вираз XPath. Якщо властивість не існує, то стиль застосовують до всіх екземплярів географічних об'єктів, вибраних атрибутом `featureType` або `baseType`.

### И.3.5 `queryGrammar`

Значення цієї властивості, що визначено як перелік, визначає граматику, яку використовують у контенті елементу `gml:featureConstraint`. Перелік допускає три значення: `"Xpath"`, `"Xquery"` та `"other"`.

Стилізація географічних об'єктів означає окремі аспекти чи аспекти географічного об'єкту. Можна стилізувати геометрію об'єкту, топологію або відображати довільний текстовий рядок. Стиль географічного об'єкту містить три дескриптори стилю для відповідних цілей: `gml:GeometryStyle`, `gml:TopologyStyle` та `gml:LabelStyle`.

## И.4 Стиль геометрії

Значення властивості `gml:geometryStyle` є дескриптором `gml:GeometryStyle`, який описує стиль для однієї геометрії географічного об'єкту. Будь-яке число дескрипторів стилю геометрії можна призначати для одного стилю географічного об'єкту. Звичайно це потрібно для географічних об'єктів із кількома геометричними властивостями.

```
<element name="geometryStyle" type="gml:GeometryStylePropertyType"/>
```

```
<complexType name="GeometryStylePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GeometryStyle" />
```

```

</sequence>
<attribute name="about" type="anyURI"/>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="GeometryStyle" type="gml:GeometryStyleType" substitutionGroup="gml:AbstractGML"/>

<complexType name="GeometryStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <choice>
          <element ref="gml:symbol"/>
          <element name="style" type="string"/>
        </choice>
        <element ref="gml:labelStyle" minOccurs="0"/>
      </sequence>
      <attribute name="geometryProperty" type="string"/>
      <attribute name="geometryType" type="string"/>
    </extension>
  </complexContent>
</complexType>

```

`gml:geometryStyle` визначений у той же спосіб, що й інші GML властивості, які дозволяють посилання на значення, віддалені чи включені.

Атрибут `gml:geometryProperty` на `gml:GeometryStyle` визначає ім'я властивості геометрії географічного об'єкту, до якого застосований цей дескриптор стилю геометрії. Потрібно визначати тип геометрії із використанням атрибуту `geometryType`, оскільки прикладна схема властивості геометрії може допускати різні геометрії як її значення.

Властивість `gml:symbol` описано в И.7.2.

Стиль властивості скасовано.

### **И.5 Стиль топології**

Значення властивості `gml:topologyStyle` є дескриптором `gml:TopologyStyle`, який описує стиль для однієї властивості топології. Аналогічно до `gml:GeometryStyle`, географічний об'єкт може мати кілька властивостей топології, тому дескриптори стилю множинних топологій можна визначати в межах одного стилю географічного об'єкту:

```

<element name="topologyStyle" type="gml:TopologyStylePropertyType"/>

<complexType name="TopologyStylePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TopologyStyle"/>
  </sequence>
  <attribute name="about" type="anyURI"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="TopologyStyle" type="gml:TopologyStyleType" substitutionGroup="gml:AbstractGML"/>

```

```

<complexType name="TopologyStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <choice>
          <element ref="gml:symbol"/>
          <element name="style" type="string"/>
        </choice>
        <element ref="gml:labelStyle" minOccurs="0"/>
      </sequence>
      <attribute name="topologyProperty" type="string"/>
      <attribute name="topologyType" type="string"/>
    </extension>
  </complexContent>
</complexType>

```

Властивість `gml:topologyStyle` визначена у той же спосіб, що й інші GML властивості, що допускають посилання на значення, включене або віддалене.

Атрибут `topologyProperty` на дескрипторі `gml:TopologyStyle` визначає ім'я властивості топології географічного об'єкту, до якого застосований цей дескриптор стилю топології. Потрібно визначити тип топології з використанням атрибуту `topologyType`, оскільки прикладна схема властивості топології може допускати різні топології як її значення.

Властивість `gml:symbol` описано в И.7.2.

*Стиль властивості скасовано.*

### **И.6 Стиль написів**

Значення властивості `gml:labelStyle` є дескриптором `gml:LabelStyle`, який описує стиль для тексту, що треба відобразити при графічному поданні географічного об'єкту. Контент напису не обов'язково визначений у GML наборі даних. Точніше, контент може бути статичним текстом, визначеним у стилі як такому, та текстом із GML набору даних.

Стиль напису має два елементи: `gml:style`, який визначає стиль, та `gml:label`, який використовують для складання контенту напису:

```

<element name="labelStyle" type="gml:LabelStylePropertyType"/>

<complexType name="LabelStylePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:LabelStyle"/>
  </sequence>
  <attribute name="about" type="anyURI"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="LabelStyle" type="gml:LabelStyleType" substitutionGroup="gml:AbstractGML"/>

<complexType name="LabelStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">

```

```

    <sequence>
      <element name="style" type="string"/>
      <element name="label" type="gml:LabelType"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

Властивість `gml:labelStyle` визначають у той же спосіб, що й інші GML властивості, які допускають посилання на значення, включене або віддалене.

Елемент `gml:style` використовують для визначення стилю рендерованого тексту. Типом цього елемента є `string`, а CSS2 граматику (Cascading Style Sheet Version 2.0) виразів стилізації використовують для виразу графічних властивостей.

*Приклад 1.* Наступний стиль географічного об'єкту показує використання елемента `gml:style` у контексті стилю геометрії:

```

<gml:FeatureStyle featureType="exp:City">
  <gml:GeometryStyle>
    <gml:style>fill:blue;stroke:white</gml:style>
  </gml:GeometryStyle>
</gml:FeatureStyle>

```

Як зауважено, властивість `gml:label` на дескрипторі `gml:LabelStyle` має текстовий контент, який можна скласти із статичного тексту та тексту, запозиченого з GML даних:

```

<complexType name="LabelType" mixed="true">
  <sequence>
    <element name="LabelExpression" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute ref="gml:transform"/>
</complexType>

```

Модель контенту є мішаною для того, щоб допускати і текстовий контент, і необмежене число елементів `gml:LabelExpression`. Значенням елемента `gml:LabelExpression` є XPath вираз, що вибирає значення деякої властивості географічного об'єкту.

*Приклад 2.* Фрагмент GML даних та відповідний стиль `gml:label`:

```

<exp:City>
  <gml:name>Belgrade</gml:name>
  <exp:size>1,700,000</exp:size>
  <gml:extentOf>
    ...
  </gml:extentOf>
</exp:City>

<gml:FeatureStyle featureType="exp:City">
  <gml:LabelStyle>
    <gml:style>font-family:Verdana;font-size:16;fill:red</gml:style>
    <gml:label>
      City:

```

```

<gml:LabelExpression>//City/name</gml:LabelExpression>
, Size:
<gml:LabelExpression>//City/size</gml:LabelExpression>
</gml:label>
</gml:LabelStyle>
</gml:FeatureStyle>

```

Результатом цього стилю напису є візуалізація такого тексту:

**City: Belgrade, Size: 1,700,000**

## **И.7 Загальні елементи стилізації**

### **И.7.1 Огляд**

Деякі загальні елементи стилізації використовують у дескрипторах множинних стилів. Елемент `gml:symbol` використовують у дескрипторах стилів геометрії та топології. Атрибути `spatialResolution`, `styleVariation` та `animation` задекларовані в `gml:BaseStyleDescriptorType`, та успадковані дескрипторами стилів геометрії, топології, написів та графів.

### **И.7.2 symbol**

Елемент властивості `gml:symbol` визначає графічний символ, який використовують для рендерінгу геометрії чи топології. Символ є описом графічних атрибутів графічного об'єкту без спеціального неявного смисла. Це може бути описом лінії, кола, полігону чи складнішого креслення. Використовуючи елемент символу, можна визначити певний символ у два способи:

- віддалений: подібно до будь-якій іншій віддаленій властивості, властивість символу має атрибути `gml:AssociationAttributeGroup`, які дозволяють визначити посилання, що вказує на віддалений об'єкт;
- включенням: значенням властивості `gml:symbol` є будь-який визначальних; це дозволяє визначити довільну граматику для символу.

Цей елемент має два додаткових атрибути: `symbolType` та `transform`. Атрибут `symbolType` є переліком та може приймати одне з трьох значень: "svg", "xpath" чи "other". Аплікації використовують значення цього атрибуту для того, щоб вирішити, як інтерпретувати символ.

Атрибут `transform` дозволяє визначити вираз трансформації, що буде застосована до символу в фазі рендерінгу. Його тип є `string`, а значення визначене в SVG (атрибут `transform`).

### **И.7.3 styleVariation**

Функція властивості `gml:styleVariation` є різноманітною:

- стилізація написів: стиль напису не має символу, асоційованого з ним, оскільки контент не є графічним, але поданий текстом; цю властивість можна використовувати для визначення атрибутів стилю;
- варіації стилізації символу: один символ часто використовують у різних випадках із незначними відмінностями; було б обтяжливим створювати та супроводжувати великий набір фактично ідентичних символів; легше створювати та використовувати тільки один символ та подавати незначні відмінності в стилі, використовуючи цю властивість;

– параметризовані стилі: параметризовані стилі – це стилі, атрибути яких залежать від деякої властивості географічного об’єкту, який стилізують.

*Приклад 1.* Місто можна стилізувати по-різному залежно від його населення. Властивість `gml:styleVariation` дозволяє визначити такі залежності.

Модель контенту цієї властивості є такою:

```
<complexType name="StyleVariationType">
  <simpleContent>
    <extension base="string">
      <attribute name="styleProperty" type="string" use="required"/>
      <attribute name="featurePropertyRange" type="string" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

Вона має два атрибути: `styleProperty` та `featurePropertyRange`. Значення `styleProperty` є іменем атрибуту стилізації SVG, таких як “stroke”, “fill”, і т. і. Воно визначає, який атрибут стилю ця властивість встановлює або перевизначає. Значення елемента `styleVariation` є значенням атрибуту стилізації, визначене через `styleProperty`. Значення може бути константним виразом або XPath виразом.

Атрибут `featurePropertyRange` визначає піднабір географічних об’єктів, до яких застосовано варіацію. Його значення є XPath виразом.

*Приклад 1.* Далі показані дві варіації стилю символу для географічного об’єкту City (місто). Цей об’єкт стилізований із використанням круглого символу. Радіус круга залежить від населення (population) міста і обрахований по-різному, залежно від того, більше чи менше населення за 2000000:

```
<gml:FeatureStyle featureType="exp:City">
  <gml:GeometryStyle>
    <gml:styleVariation
      styleProperty="r"
      featurePropertyRange="population &gt;= 2000000">population div 1000000</gml:styleVariation>
    <gml:styleVariation
      styleProperty="r"
      featurePropertyRange="population &lt; 2000000">population div 1000000</gml:styleVariation>
    <gml:symbol xlink:href="http://www.opengis.org/symbols/City.xml#City"/>
  </gml:GeometryStyle>
</gml:FeatureStyle>
```

#### **И.7.4 spatialResolution**

Значення елемента властивості `gml:spatialResolution` є `gml:MeasureType`. У GML стилізації за умовчанням смисл цього елемента базується на відповідному визначенні в ISO 19115, де він визначений як коефіцієнт, що надає загальне розуміння щільності просторових даних у наборі даних. GML не визначає точне використання цього атрибуту інакше, ніж у цьому неформальному визначенні. Розробники аплікацій можуть використовувати `spatialResolution` в різні способи.

*Приклад 1.* Це можна використовувати як знаменник масштабу карти (1:50000, 1:25000, і т. і.).



*Приклад 2.* В аплікаціях можна також використовувати його значення для визначення, як треба відображати географічні об'єкти за різних масштабів. Наприклад, місто та його об'єкти звичайно показані детальніше на карті великого масштабу, і, можливо, тільки як один умовний знак на карті малого масштабу, або берегова лінія може бути нанесена в деталях на карті великого масштабу, тоді як на карті малого масштабу можна пропустити деякі координати для кращої продуктивності.

### **И.7.5 animation**

Атрибути анімації використовують для опису анімаційної поведінки геометрії, топології, написів чи графів.

Ці атрибути визначені в W3C SMIL (SMIL 2.0 BasicAnimation Elements), див. таблицю И.1.

**Таблиця И.1** – Атрибути, які використовують для анімації

<b>Атрибут</b>	<b>Функція використання</b>
animate	Загальний атрибут анімації
animateMotion	Переміщення елемента по шляху
animateColor	Анімація атрибутів кольору
set	Встановлення значення атрибуту для визначеної тривалості

### **И.8 Стиль графів**

Властивість `gml:graphStyle` дескриптору `gml:FeatureStyle` має в якості свого значення дескриптор `gml:GraphStyle`, що визначає атрибути стилю графу, сформованого набором географічних об'єктів. Визначення властивості стилю графу та дескриптору показані далі:

```
<element name="graphStyle" type="gml:GraphStylePropertyType"/>

<complexType name="GraphStylePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GraphStyle"/>
  </sequence>
  <attribute name="about" type="anyURI"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="GraphStyle" type="gml:GraphStyleType" substitutionGroup="gml:AbstractGML"/>

<complexType name="GraphStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <element name="planar" type="boolean" minOccurs="0"/>
        <element name="directed" type="boolean" minOccurs="0"/>
        <element name="grid" type="boolean" minOccurs="0"/>
        <element name="minDistance" type="double" minOccurs="0"/>
        <element name="minAngle" type="double" minOccurs="0"/>
        <element name="graphType" type="gml:GraphTypeType" minOccurs="0"/>
        <element name="drawingType" type="gml:DrawingTypeType" minOccurs="0"/>
        <element name="lineType" type="gml:LineTypeType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        <element name="aestheticCriteria" type="gml:AesheticCriteriaType"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</extension>
</complexContent>
</complexType>

<simpleType name="GraphTypeType">
    <restriction base="string">
        <enumeration value="TREE"/>
        <enumeration value="BICONNECTED"/>
    </restriction>
</simpleType>

<simpleType name="DrawingTypeType">
    <restriction base="string">
        <enumeration value="POLYLINE"/>
        <enumeration value="ORTHOGONAL"/>
    </restriction>
</simpleType>

<simpleType name="LineTypeType">
    <restriction base="string">
        <enumeration value="STRAIGHT"/>
        <enumeration value="BENT"/>
    </restriction>
</simpleType>

<simpleType name="AesheticCriteriaType">
    <restriction base="string">
        <enumeration value="MIN_CROSSINGS"/>
        <enumeration value="MIN_AREA"/>
        <enumeration value="MIN_BENDS"/>
        <enumeration value="MAX_BENDS"/>
        <enumeration value="UNIFORM_BENDS"/>
        <enumeration value="MIN_SLOPES"/>
        <enumeration value="MIN_EDGE_LENGTH"/>
        <enumeration value="MAX_EDGE_LENGTH"/>
        <enumeration value="UNIFORM_EDGE_LENGTH"/>
        <enumeration value="MAX_ANGULAR_RESOLUTION"/>
        <enumeration value="MIN_ASPECT_RATIO"/>
        <enumeration value="MAX_SYMMETRIES"/>
    </restriction>
</simpleType>

```

Властивість `gml:graphStyle` визначена у той же самий спосіб, що й інші GML властивості, які дозволяють посилання на значення, віддалене чи включене.

Дескриптор стилю графу описує стиль для графу як цілого, а не для окремих елементів графу. Він успадковує від базової моделі контенту загальні властивості стилізації, описані в підрозділі И.7.

Цей дескриптор додає до базової моделі контенту групу властивостей, специфічних для стилізації графу – вони описують граф у термінах його специфічних характеристик. Ці властивості описані в таблиці И.2.

**Таблиця И.2** – Елементи, які використовують у стилізації графу

Елемент	Тип	Використання
planar	Булевський	Якщо істина, то ребра графу не перетинаються (планарний граф); в іншому разі вони можуть перетинатися
directed	Булевський	Якщо істина, то граф спрямований; в іншому разі він не є спрямований
grid	Булевський	Якщо істина, то координати вузлів, перетинів та поворотів мають цілочислові значення; в іншому разі вони можуть мати десяткові значення
minDistance	Довге дійсне	Рекомендація з мінімальної віддалі між вершинами та incident ребрами
minAngle	Довге дійсне	Рекомендація з мінімального куту між послідовними incident ребрами (кутова роздільна здатність)
graphType	Перелік	Значення може бути TREE або BICONNECTED
drawingType	Перелік	Тип креслення з урахуванням ортогональності ребер; значення може бути POLYLINE або ORTHOGONAL
lineType	Перелік	Визначає, чи будуть будь-які непрямолінійні ребра; значення може бути STRAIGHT або BENT
aestheticCriteria	Перелік	Рекомендація з загального контуру графу відповідно до індивідуальних естетичних критеріїв; значення може бути одним із наступних: MIN_CROSSINGS, MIN_AREA, MIN_BENDS, MAX_BENDS, UNIFORM_BENDS, MIN_SLOPES, MIN_EDGE_LENGTH, MAX_EDGE_LENGTH, UNIFORM_EDGE_LENGTH, MAX_ANGULAR_RESOLUTION, MIN_ASPECT_RATIO або MAX_SYMMETRIES

ДОДАТОК К  
(довідковий)

**ЗВОРОТНА СУМІСНІСТЬ З ПОПЕРЕДНІМИ ВЕРСІЯМИ GML**

**К.1 Огляд**

У цьому додатку визначені скасовані компоненти схеми та їх вимоги (див. підрозділ 5.3)

**К.2 Базові компоненти схеми**

**К.2.1 remoteSchema**

Атрибут *remoteSchema* був призначений для указання схеми, що обмежує опис віддаленого ресурсу, на який спрямоване *xlink* посилання. Використання цього атрибуту було скасоване, натомість із цією метою можна використовувати *xlink:role* (див. підрозділ 8.1).

```
<attribute name="remoteSchema" type="anyURI"/>
```

**К.2.2 member**

Конкретний елемент властивості, що називали “member”, був раніше задекларований таким чином:

```
<element name="member" type="gml:AssociationRoleType"/>
```

Натомість треба використовувати елементи властивості, визначені в прикладній схемі.

**К.2.3 ArrayAssociationType**

Для властивості, яку можна закодувати тільки як включену, шаблон типу властивості закодовано таким чином:

```
<complexType name="ArrayAssociationType">
  <sequence>
    <element ref="gml:AbstractObject" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Цей тип було замінено на типи, похідні від *gml:AbstractMemberType* (див. підпункт 7.2.5.1)

**К.2.4 members**

Конкретний елемент властивості, що називають “members”, раніше був задекларований як

```
<element name="members" type="gml:ArrayAssociationType"/>
```

Натомість треба використовувати елементи властивості, визначені в прикладній схемі,

**К.2.5 featureProperty, featureMember, featureMembers**

Конкретні елементи *gml:featureMember* та *gml:featureProperty* використовували шаблон *gml:AssociationRoleType* у своїй моделі контенту та були задекларовані таким чином:

```
<element name="featureMember" type="gml:FeaturePropertyType"/>
<element name="featureProperty" type="gml:FeaturePropertyType"/>
```

Конкретні елементи *gml:featureMembers* містять масив географічних об'єктів та задекларовані таким чином:

```
<element name="featureMembers" type="gml:FeatureArrayPropertyType"/>
```

Ці елементи замінено на елементи, визначені в прикладних схемах.

### К.2.6 StringOrRefType

*gml:StringOrRefType* – це тип, наданий для того, щоб містити розширені текстові значення. Він задекларований таким чином:

```
<complexType name="StringOrRefType">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </simpleContent>
</complexType>
```

Використання віддалених посилань на цей тип скасовано. Цей тип був раніше наявний всюди, де була потрібна властивість «текстового» типу. Вона була в форматі рядку, тому текст можна включити, але на значення також могло бути спрямоване посилання віддалено через атрибут *xlink:href*. Якщо було присутнє віддалене посилання, то значення, отримане через проходження посиланням, вважали за її значення.

Для посилання на віддалене рядкове значення натомість треба використовувати атрибут *xlink:href* елемента типу *gml:ReferenceType* (див. підпункт 7.2.3.7), тобто, *gml:descriptionReference*.

### К.2.7 Array, ArrayType, Bag, BagType

Дві конкретні колекції об'єктів були надані, але скасовані. Натомість GML колекції об'єктів треба будувати в прикладній схемі, як визначено в пункті 7.2.5.

*gml:Bag* був призначений для загальних колекцій без жодних припущень щодо типу, порядку чи унікальності об'єктів, що є її членами:

```
<element name="Bag" type="gml:BagType" substitutionGroup="gml:AbstractGML"/>
<complexType name="BagType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:member" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:members" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*gml:Array* був призначений для використання для колекції, об'єкти-члени якої однорідного типу, та в якій їх порядок є суттєвим:

```
<element name="Array" type="gml:ArrayType" substitutionGroup="gml:AbstractGML"/>
```

```

<complexType name="ArrayType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:members" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

### **K.2.8 metaDataProperty, MetaDataPropertyType, AbstractMetaData, AbstractMetaDataType**

Компоненти схеми, визначені в цьому підрозділі, замінено на компоненти схеми, визначені в пункті 7.2.6

Ця властивість містить пакет метаданих, що містить властивості метаданих у кодуванні, використаному в попередній версії GML, або посилається на нього. Цей елемент було скасовано та замінено елементами, модель контенту яких отримана від *gml:AbstractMetadadataPropertyType*. Більш докладно див. пункт 7.2.6.

```

<element name="metaDataProperty" type="gml:MetaDataPropertyType"/>
<complexType name="MetaDataPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractMetaData"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attribute name="about" type="anyURI"/>
</complexType>

```

Необов'язковий атрибут “*about*” надає URI, що вказує на абстрактний елемент чи діапазон елементів, або інший ресурс, на який посилаються ці метадані.

Значення *metaDataProperty* є абстрактним елементом *gml:AbstractMetaData*, що діє як вмістилище для «будь-якого пакету властивостей метаданих», визначений таким чином:

```

<element name="AbstractMetaData" type="gml:AbstractMetaDataType" abstract="true"
  substitutionGroup="gml:AbstractObject"/>

<complexType name="AbstractMetaDataType" abstract="true" mixed="true">
  <sequence/>
  <attribute ref="gml:id"/>
</complexType>

```

### **K.2.9 GenericMetaData, GenericMetaDataType**

Для зручності загальний конкретний елемент *MetaData* був наданий у попередній версії GML. Цей елемент скасовано та замінено компонентами схеми, визначеними в пункті 7.2.6.

```

<element name="GenericMetaData" type="gml:GenericMetaDataType"
  substitutionGroup="gml:AbstractMetaData"/>

<complexType name="GenericMetaDataType" mixed="true">

```

```

<complexContent mixed="true">
  <extension base="gml:AbstractMetaDataType">
    <sequence>
      <any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

### К.3 Basic types, Null

gml:Null замінено атрибутами nillable та nilReason, як визначено в підпункті 8.2.3.2.

Елемент gml:Null задекларований таким чином

```
<element name="Null" type="gml:NilReasonType"/>
```

Приклад 1. Цей елемент може зустрічатися в документах екземпляру даних у такому вигляді:

```
<gml:Null>withheld</gml:Null>
```

```
<gml:Null>http://my.big.org/explanations/theDogAteIt</gml:Null>
```

У першому прикладі використані включені значення для Null. У другому прикладі міститься посилання на пояснення, наявне десь іще, ідентифіковане з допомогою URI..

Метою надання елементу *gml:Null* є наступне: для того, щоб будувати модель контенту, де можна пропустити значення, можна було б використовувати обмеження потужності множини в XML схемі з використанням конструкції *minOccurs="0"*. Однак, цей підхід являє ризик того, що підставу для відсутності значення можна неправильно інтерпретувати. Як альтернатива, можна включити елемент *gml:Null* як член групи вибору *choice*, разом із елементом типу даних “нормального значення”.

Приклад 1. Модель контенту, описана фрагментом схеми

```

<element name="footprint">
  <complexType>
    <choice>
      <element ref="gml:Envelope"/>
      <element ref="gml:Null"/>
    </choice>
  </complexType>
</element>

```

дозволяє один з таких дійсних екземплярів даних

```

<footprint>
  <gml:Envelope> ... </gml:Envelope>
</footprint>

```

```

<footprint>
  <gml:Null>inapplicable</gml:Null>
</footprint>

```

Це дозволяє появу гіпотетичного елемента “footprint” в документі екземпляру, необов’язково містячи явне маркерне указання причини відсутності у нього значення замість наявності семантики, передбаченою відсутністю значення.

#### К. 4 Географічні об’єкти

##### К.4.1 *location*, *LocationPropertyType*, *LocationKeyWord*, *LocationString*

Елемент *gml:location* є властивістю зручності, що описувала згенероване місцеположення географічного об’єкту.

```
<element name="location" type="gml:LocationPropertyType"/>

<complexType name="LocationPropertyType">
<sequence >
  <choice>
    <element ref="gml:AbstractGeometry"/>
    <element ref="gml:LocationKeyWord"/>
    <element ref="gml:LocationString"/>
    <element ref="gml:Null"/>
  </choice>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Значення місцеположення може бути геометрією, рядком місцеположення, ключовим словом місцеположення або нулем.

*gml:location* та *gml:LocationPropertyType* скасовані.

**Примітка 1.** Гнучка модель контенту місцеположення показала труднощі практичної реалізації.

Рядок місцеположення – це текст, що має описувати місцеположення. Воно описане таким чином:

```
<element name="LocationString" type="gml:StringOrRefType"/>
```

Ключове слово *gml:LocationKeyWord* скасовано та замінено на *gml:locationName*.

Ключове слово є кодом, звичайно вибраним із контрольованого списку. Воно задеклароване таким чином:

```
<element name="LocationKeyWord" type="gml:CodeType"/>
```

*gml:LocationString* скасовно та замінено *gml:locationReference* та *gml:locationName* (див. пункт 9.4.2).

##### К.4.2 *priorityLocation*, *priorityLocationType*

Властивість *gml:priorityLocation* була надана для розробників GML прикладної схеми, які бажають надавати пріоритетні місцеположення для їх географічних об’єктів. *gml:priorityLocation* має таку модель контенту:

```
<element name="priorityLocation" type="gml:PriorityLocationPropertyType"
substitutionGroup="gml:location"/>

<complexType name="PriorityLocationPropertyType">
  <complexContent>
    <extension base="gml:LocationPropertyType">
```



```

    <attribute name="priority" type="string" />
  </extension>
</complexContent>
</complexType>

```

Треба зауважити, що це просто додає рядок пріоритету до базової властивості *gml:location* для призначення рівнів важливості різним місцезположенням.

#### К.4.3 BoundedFeatureType

Просте обмеження на тип *gml:AbstractFeatureType* було раніше запропоновано для надання обов'язковості необов'язковій властивості *boundedBy*. *gml:BoundedFeatureType* було визначено таким чином

```

<complexType name="BoundedFeatureType" abstract="true">
  <complexContent>
    <restriction base="gml:AbstractFeatureType">
      <sequence>
        <group ref="gml:StandardObjectProperties"/>
        <element ref="gml:boundedBy"/>
        <element ref="gml:location" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>

```

#### К.4.4 AbstractFeatureCollectionType, AbstractFeatureCollection, FeatureCollection, FeatureCollectionType

GML колекції географічних об'єктів у попередніх версіях GML були отримані шляхом розширення чи обмеження від *gml:AbstractFeatureCollectionType*, що визначено таким чином:

```

<complexType name="AbstractFeatureCollectionType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:featureMember" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:featureMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Властивість *gml:featureMember* (але не властивість *gml:featureMembers*) дотримано згідно до шаблону асоціації, і таким чином вона посилається на "віддалений" географічний об'єкт засобами атрибуту *xlink:ref*.

Композитна властивість *gml:featureMembers* включає набір членів колекції географічних об'єктів назалежно від їх семантичного типу як географічних об'єктів. *gml:featureMember* включає одиночний екземпляр географічного об'єкту. Властивості *gml:featureMember* та *gml:featureMembers* можуть зустрічатися в тій же самій колекції географічних об'єктів, але там може бути тільки одна властивість *gml:featureMembers*.

GML колекції географічних об'єктів є самі по собі дійсними GML географічними об'єктами та можуть мати *gml:location* та інші властивості, як визначено в їх GML прикладній схемі.

```
<element name="AbstractFeatureCollection" type="gml:AbstractFeatureCollection
Type" abstract="true"substitutionGroup="gml:AbstractFeature"/>
```

Абстрактний елемент *gml:AbstractFeatureCollection* виступає головою групи заміни, яка може містити будь-які елементи, модель контенту яких отримано від *gml:AbstractFeatureType*. Це можна використовувати в побудові моделей контенту.

Схема також надає колекцію конкретних географічних об'єктів:

```
<element name="FeatureCollection" type="gml:FeatureCollectionType" substitutionGroup="gml:AbstractFeature"/>
<complexType name="FeatureCollectionType">
  <complexContent>
    <extension base="gml:AbstractFeatureCollectionType"/>
  </complexContent>
</complexType>
```

Користувачі конкретної *gml:FeatureCollection* мають пам'ятати, що вона дозволяє тільки дійсний GML географічний об'єкт у якості члена.

Модель контенту GML колекції об'єктів у попередніх версіях GML була отримана від *gml:AbstractFeatureCollectionType*. Це, в свою чергу, походить від *gml:AbstractFeatureType*. Тому колекції географічних об'єктів є географічними об'єктами, і взагалі можуть замінювати *gml:AbstractFeature*.

Компоненти схеми, визначені в цьому підрозділі, скасовані та замінені правилами для GML колекцій об'єктів, визначеними в пункті 9.9.1.

#### **К.4.5 Просторові властивості**

Взагалі за визначення властивостей географічного об'єкту відповідає розробник прикладної схеми. GML було раніше визначено як набір наперед визначених елементів просторової властивості для асоціювання екземплярів цих просторових типів із географічними об'єктами. Це було скасовано, натомість треба використовувати імена властивостей, притаманні для даної прикладної схеми:

– Описові імена, що надають набір імен властивостей, часто використовуваних у прикладних схемах. Ними є:

```
<element name="centerOf" type="gml:PointPropertyType"/>
<element name="position" type="gml:PointPropertyType"/>
<element name="extentOf" type="gml:SurfacePropertyType"/>
<element name="edgeOf" type="gml:CurvePropertyType"/>
<element name="centerLineOf" type="gml:CurvePropertyType"/>

<element name="multiLocation" type="gml:MultiPointPropertyType"/>
<element name="multiCenterOf" type="gml:MultiPointPropertyType"/>
<element name="multiPosition" type="gml:MultiPointPropertyType"/>
<element name="multiCenterLineOf" type="gml:MultiCurvePropertyType"/>
<element name="multiEdgeOf" type="gml:MultiCurvePropertyType"/>
<element name="multiCoverage" type="gml:MultiSurfacePropertyType"/>
```

```
<element name="multiExtentOf" type="gml:MultiSurfacePropertyType"/>
```

Ці елементи властивості забезпечують спільні імена ролей для геометрії географічних об'єктів. Однак, специфічні семантики цих імен ролей не визначені.

– Формальні імена, що позначають просторові властивості у спосіб, базований на типі геометрії чи топології, дозволеної в якості значення властивості. Ці імена базуються на імені просторового типу із суфіксом "Property". Ці типи властивості звичайно визначені для користування в самій GML схемі. Їх не слід використовувати для елементів властивості в прикладних схемах. Всі формальні імена, які не використовують у GML схемі, скасовані:

```
<element name="topoComplexProperty" type="gml:TopoComplexPropertyType"/>
<element name="multiPointProperty" type="gml:MultiPointPropertyType"/>
<element name="multiCurveProperty" type="gml:MultiCurvePropertyType"/>
<element name="multiSurfaceProperty" type="gml:MultiSurfacePropertyType"/>
<element name="multiSolidProperty" type="gml:MultiSolidPropertyType"/>
<element name="multiGeometryProperty" type="gml:MultiGeometryPropertyType"/>
<element name="pointArrayProperty" type="gml:PointArrayPropertyType"/>
<element name="curveArrayProperty" type="gml:CurveArrayPropertyType"/>
<element name="surfaceArrayProperty" type="gml:SurfaceArrayPropertyType"/>
<element name="solidArrayProperty" type="gml:SolidArrayPropertyType"/>
```

Семантику спеціально для цих імен ролей (тобто, "Що означає *multiPointProperty* об'єкту?") не визначено.

## **К.5 Координатна геометрія, геометричні примітиви**

### **К.5.1 coordinates**

```
<element name="coordinates" type="gml:CoordinatesType" />
```

Елемент *gml:coordinates* скасовано та замінено на *gml:posList* (див. підпункт 10.1.4.2).

### **К.5.2 pos у EnvelopeType**

Натомість треба використовувати властивості *gml:lowerCorner* та *gml:upperCorner* у *gml:EnvelopeType* (див. пункт 10.1.4.6).

### **К.5.3 pointRep**

```
<element name="pointRep" type="gml:PointPropertyType"/>
```

Елемент властивості скасовано. Натомість треба використовувати *gml:pointProperty* (див. пункт 10.3.2).

### **К.5.4 polygonPatches**

```
<element name="polygonPatches" type="gml:SurfacePatchArrayPropertyType"
substitutionGroup="gml:patches"/>
```

*gml:polygonPatches* інкапсулюють ділянки полігону потенційної поверхні. Натомість треба використовувати *gml:patches*.

### **К.5.5 trianglePatches**

```
<element name="trianglePatches" type="gml:SurfacePatchArrayPropertyType"
substitutionGroup="gml:patches"/>
```

*gml:trianglePatches* інкапсулюють ділянки трикутників нерегулярної мережі трикутників. Натомість треба використовувати *gml:patches*.  
*instead.*

## К.6 Референційні системи координат

### К.6.1 baseGeographicCRS

```
<element name="baseGeographicCRS" type="gml:GeographicCRSPropertyType"/>
```

*gml:baseGeographicCRS* є роллю в асоціації географічної референційної системи координат, яка використана в цій проєційованій референційній системі координат. (див. 12.3.3.15).

### К.6.2 GeographicCRS

```
<element name="GeographicCRS" type="gml:GeographicCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="GeographicCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
      <sequence>
        <element ref="gml:usesEllipsoidalCS"/>
        <element ref="gml:usesGeodeticDatum"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*gml:GeographicCRS* є референційною системою координат, що базується на еліпсоїдній апроксимації геоїду; це забезпечує точне подання геометрії географічних об'єктів для значної частини земної поверхні. *gml:GeographicCRS* скасовано та замінено на *gml:GeodeticCRS* (див. підпункт 12.3.3.4).

```
<complexType name="GeographicCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GeographicCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

*gml:GeographicCRSPropertyType* є типом властивості для ролей в асоціації до географічної референційної системи координат, що або містить визначення цієї референційної системи, або посилається на нього. Тип властивості скасовано та замінено на *gml:GeodeticCRSPropertyType* (див. 12.3.3.9).

### К.6.3 GeocentricCRS

```
<element name="GeocentricCRS" type="gml:GeocentricCRSType"
substitutionGroup="gml:AbstractSingleCRS"/>
```

```
<complexType name="GeocentricCRSType">
  <complexContent>
    <extension base="gml:AbstractCRSType">
```

```

<sequence>
  <choice>
    <element ref="gml:usesCartesianCS"/>
    <element ref="gml:usesSphericalCS"/>
  </choice>
  <element ref="gml:usesGeodeticDatum"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

*gml:GeocentricCRS* є трьохвимірною референчною системою з початком у приблизному центрі маси Землі. Геоцентрична РСК ураховує кривизну Землі за рахунок її подання в трьохвимірному просторовому вигляді.

*gml:GeocentricCRS* скасовано та замінено на *gml:GeodeticCRS* (див. підпункт 12.3.3.4).

```

<complexType name="GeocentricCRSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:GeocentricCRS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

*gml:GeocentricCRSPropertyType* є типом властивості для ролей в асоціації до геодезичної референчної системи координат. Тип властивості був скасований та замінений на *gml:GeodeticCRSPropertyType* (див. підпункт 12.3.3.9).

#### **К.6.4 uom**

```

<attribute name="uom" type="anyURI"/>

```

Атрибут *uom* надає ідентифікатор одиниці вимірювання, та скасований. Натомість треба використовувати локальні атрибути *uom*.

#### **К.6.5 ObliqueCartesianCS**

```

<element name="ObliqueCartesianCS" type="gml:ObliqueCartesianCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>

```

```

<complexType name="ObliqueCartesianCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>

```

*gml:ObliqueCartesianCS* є двох- чи трьохвимірною системою координат із прямими осями, але не обов'язково ортогональними. *ObliqueCartesianCS* має мати дві чи три асоціації *gml:usesAxis*. Ці елемент та тип було скасовано та замінено на *gml:AffineCS* та його тип (див. підпункт 12.4.4.19).

```

<complexType name="ObliqueCartesianCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:ObliqueCartesianCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

*gml:ObliqueCartesianCSPropertyType* є типом властивості для ролей в асоціації до косої декартової системи координат, який або містить визначення цієї системи координат, або посилається на нього. Цей тип властивості скасовано та замінено на *gml:AffineCSPropertyType*.

### К.6.6 TemporalCS

```

<element name="TemporalCS" type="gml:TemporalCSType"
substitutionGroup="gml:AbstractCoordinateSystem"/>
<complexType name="TemporalCSType">
  <complexContent>
    <extension base="gml:AbstractCoordinateSystemType"/>
  </complexContent>
</complexType>

```

*gml:TemporalCS* є одновимірною системою координат, що містить одну часову вісь, яку використовують для опису часової позиції точки в визначених одиницях часу відносно визначеного початку часу. *TemporalCS* повинна мати один елемент властивості *gml:usesAxis*. Ці елемент та тип скасовано та замінено на *gml:TimeCS* та його тип (див. підпункт 12.4.4.7).

```

<complexType name="TemporalCSPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:TemporalCS"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

*gml:TemporalCSPropertyType* є типом властивості для ролей в асоціації до часової системи координат, що або містить визначення цієї системи координат, або посилається на неї. Цей тип властивості скасовано.

### К.6.7 greenwichLongitude

Використання скасованого *gml:AngleChoiceType* у *gml:greenwichLongitude* (див. пілпункт 12.5.3.7) усунено.

### К.6.8 AbstractOperation

```

<element name="AbstractOperation" type="gml:AbstractCoordinateOperationType" abstract="true"
substitutionGroup="gml:AbstractSingleOperation"/>

```

*gml:AbstractOperation* є параметризованою математичною операцією на координатах, що трансформує чи перетворює координати до іншої референтної системи координат. Ця координатна операція використовує метод операції, звичайно з асоційованими значеннями параметру. Однак, методи операції та значення параметрів безпосередньо асоційовані з конкретними підтипами, але не з абстрактним типом. GML прикладна схема не може розширювати чи обмежувати цей абстрактний *complexType*. Цей елемент скасовано та замінено на *gml:AbstractSingleOperation* (див. підпункт 12.6.2.7):

```

<complexType name="OperationPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:AbstractOperation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

*gml:OperationPropertyType* є типом властивості для ролей в асоціації до абстрактної операції, який або містить визначення цієї операції, або посилається на нього. Цей тип властивості скасовано та замінено на *gml:AbstractSingleOperationPropertyType* (див. підпункт 12.6.2.8).

### К.6.9 dmsAngleValue

```
<element name="dmsAngleValue" type="gml:DMSAngleType"/>
```

*gml:dmsAngleValue* є значенням параметру кутової операції, або в форматі градус-хвилина-секунда, або в форматі одного значення. Цей елемент властивості скасовано.

### К.6.10 Перейменовані елементи властивості

Попередні версії GML містили елементи властивості, які використовували домовленість із найменування, відмінну від рекомендованої ISO/TS 19103 та використовувалися в інших місцях у GML, або через те, що ім'я властивості в концепції, що лежала в основі, змінилося. Ці елементи властивості скасовано, і натомість треба використовувати елементи властивості, згадані в групі заміни..

```

<element name="methodFormula" type="gml:CodeType" substitutionGroup="gml:formula"/>
<element name="anchorPoint" type="gml:CodeType" substitutionGroup="gml:anchorDefinition"/>
<element name="generalOperationParameter" type="gml:AbstractGeneralOperationParameterPropertyType"
  substitutionGroup="gml:parameter"/>
<element name="valueOfParameter" type="gml:OperationParameterPropertyType"
  substitutionGroup="gml:operationParameter"/>
<element name="valuesOfGroup" type="gml:OperationParameterGroupPropertyType"
  substitutionGroup="gml:group"/>
<element name="includesParameter" type="gml:AbstractGeneralOperationParameterPropertyType"
  substitutionGroup="gml:parameter"/>
<element name="definedByConversion" type="gml:GeneralConversionPropertyType"
  substitutionGroup="gml:conversion"/>
<element name="includesSingleCRS" type="gml:SingleCRSPropertyType"
  substitutionGroup="gml:componentReferenceSystem"/>
<element name="usesEllipsoidalCS" type="gml:EllipsoidalCSPropertyType"
  substitutionGroup="gml:ellipsoidalCS"/>
<element name="usesCartesianCS" type="gml:CartesianCSPropertyType" substitutionGroup="gml:cartesianCS"/>
<element name="usesSphericalCS" type="gml:SphericalCSPropertyType" substitutionGroup="gml:sphericalCS"/>
<element name="usesGeodeticDatum" type="gml:GeodeticDatumPropertyType"
  substitutionGroup="gml:geodeticDatum"/>
<element name="usesVerticalCS" type="gml:VerticalCSPropertyType"
  substitutionGroup="gml:verticalCS"/>
<element name="usesVerticalDatum" type="gml:VerticalDatumPropertyType"
  substitutionGroup="gml:verticalDatum"/>
<element name="usesCS" type="gml:CoordinateSystemPropertyType" substitutionGroup="gml:coordinateSystem"/>
<element name="usesEngineeringDatum" type="gml:EngineeringDatumPropertyType"

```

```

    substitutionGroup="gml:engineeringDatum"/>
<element name="usesAffineCS" type="gml:AffineCSPropertyType"
    substitutionGroup="gml:affineCS"/>
<element name="usesImageDatum" type="gml:ImageDatumPropertyType"
    substitutionGroup="gml:imageDatum"/>
<element name="usesObliqueCartesianCS" type="gml:ObliqueCartesianCSPropertyType"/>
<element name="usesTimeCS" type="gml:TimeCSPropertyType" substitutionGroup="gml:timeCS"/>
<element name="usesTemporalCS" type="gml:TemporalCSPropertyType"/> натомість уживати gml:timeCS )
<element name="usesTemporalDatum" type="gml:TemporalDatumPropertyType"
    substitutionGroup="gml:temporalDatum"/>
<element name="usesAxis" type="gml:CoordinateSystemAxisPropertyType" substitutionGroup="gml:axis"/>
<element name="usesPrimeMeridian" type="gml:PrimeMeridianPropertyType"
    substitutionGroup="gml:primeMeridian"/>
<element name="usesEllipsoid" type="gml:EllipsoidPropertyType" substitutionGroup="gml:ellipsoid"/>
<element name="usesSingleOperation" type="gml:CoordinateOperationPropertyType"
    substitutionGroup="gml:coordOperation"/>
<element name="usesOperation" type="gml:CoordinateOperationPropertyType"
    substitutionGroup="gml:coordOperation"/>
<element name="usesMethod" type="gml:OperationMethodPropertyType" substitutionGroup="gml:method"/>
<element name="usesValue" type="gml:AbstractGeneralParameterValuePropertyType"
    substitutionGroup="gml:parameterValue"/>
<element name="usesParameter" type="gml:AbstractGeneralOperationParameterPropertyType"
    substitutionGroup="gml:generalOperationParameter"/>
<element name="includesValue" type="gml:AbstractGeneralParameterValuePropertyType"
    substitutionGroup="gml:parameterValue"/>

```

#### К.6.11 Ref елементи властивості

Попередні версії GML містили наперед визначені елементи властивості в документах схеми референційної системи координат. Ці елементи властивості скасовано, і треба визначати елементи властивості в прикладній схемі.

```

<element name="crsRef" type="gml:CRSPropertyType"/>
<element name="singleCRSRef" type="gml:SingleCRSPropertyType"/>
<element name="compoundCRSRef" type="gml:CompoundCRSPropertyType"/>
<element name="verticalCRSRef" type="gml:VerticalCRSPropertyType"/>
<element name="projectedCRSRef" type="gml:ProjectedCRSPropertyType"/>
<element name="derivedCRSRef" type="gml:DerivedCRSPropertyType"/>
<element name="engineeringCRSRef" type="gml:EngineeringCRSPropertyType"/>
<element name="temporalCRSRef" type="gml:TemporalCRSPropertyType"/>
<element name="imageCRSRef" type="gml:ImageCRSPropertyType"/>
<element name="geocentricCRSRef" type="gml:GeocentricCRSPropertyType"/>
<element name="coordinateSystemAxisRef" type="gml:CoordinateSystemAxisPropertyType"/>
<element name="coordinateSystemRef" type="gml:CoordinateSystemPropertyType"/>
<element name="ellipsoidalCSRef" type="gml:EllipsoidalCSPropertyType"/>
<element name="cartesianCSRef" type="gml:CartesianCSPropertyType"/>
<element name="temporalCSRef" type="gml:TemporalCSPropertyType"/>
<element name="linearCSRef" type="gml:LinearCSPropertyType"/>
<element name="userDefinedCSRef" type="gml:UserDefinedCSPropertyType"/>
<element name="sphericalCSRef" type="gml:SphericalCSPropertyType"/>

```



```

<element name="polarCSRef" type="gml:PolarCSPropertyType"/>
<element name="cylindricalCSRef" type="gml:CylindricalCSPropertyType"/>
<element name="datumRef" type="gml:DatumPropertyType"/>
<element name="geodeticDatumRef" type="gml:GeodeticDatumPropertyType"/>
<element name="ellipsoidRef" type="gml:EllipsoidPropertyType"/>
<element name="primeMeridianRef" type="gml:PrimeMeridianPropertyType"/>
<element name="engineeringDatumRef" type="gml:EngineeringDatumPropertyType"/>
<element name="imageDatumRef" type="gml:ImageDatumPropertyType"/>
<element name="verticalDatumRef" type="gml:VerticalDatumPropertyType"/>
<element name="temporalDatumRef" type="gml:TemporalDatumPropertyType"/>
<element name="coordinateOperationRef" type="gml:CoordinateOperationPropertyType"/>
<element name="singleOperationRef" type="gml:SingleOperationPropertyType"/>
<element name="referenceSystemRef" type="gml:CRSPropertyType"/>
<element name="generalConversionRef" type="gml:GeneralConversionPropertyType"/>
<element name="generalTransformationRef" type="gml:GeneralTransformationPropertyType"/>
<element name="concatenatedOperationRef" type="gml:ConcatenatedOperationPropertyType"/>
<element name="passThroughOperationRef" type="gml:PassThroughOperationPropertyType"/>
<element name="conversionRef" type="gml:ConversionPropertyType"/>
<element name="transformationRef" type="gml:TransformationPropertyType"/>
<element name="abstractGeneralOperationParameterRef"
  type="gml:AbstractGeneralOperationParameterPropertyType"/>
<element name="operationParameterGroupRef" type="gml:OperationParameterPropertyType"/>
<element name="operationParameterRef" type="gml:OperationParameterPropertyType"/>
<element name="operationMethodRef" type="gml:OperationMethodPropertyType"/>
<element name="obliqueCartesianCSRef" type="gml:ObliqueCartesianCSPropertyType"/>
<element name="verticalCSRef" type="gml:VerticalCSPropertyType"/>
<element name="operationRef" type="gml:OperationPropertyType"/>

```

## К.7 Часова інформація та динамічні об'єкти

### К.7.1 SuccessionType

Часовий нелінійний граф є мережею, що складається з часових ребер. Це надає основу для описування послідовностей екземплярів географічного об'єкту або значень властивості географічного об'єкту, описаних з допомогою “заміни”, “поділу”, “злиття” та “ініціації”. Для підтримки прикладних схем, які вимагають цієї можливості, у попередніх версіях було надано *gml:SuccessionType*, щоб гарантувати узгоджену термінологію. Однак, оскільки це було ізольованою концепцією, тип скасовано. Він визначений таким чином:

```

<simpleType name="SuccessionType">
  <restriction base="string">
    <enumeration value="substitution"/>
    <enumeration value="division"/>
    <enumeration value="fusion"/>
    <enumeration value="initiation"/>
  </restriction>
</simpleType>

```

### К.7.2 MovingObjectStatus

*gml:MovingObjectStatus* є одним із прикладів, як можна розширювати *gml:AbstractTimeSlice*. Цей елемент забезпечує метод запису статусу рухомого об'єкту. Він задекларований таким чином:

```
<element name="MovingObjectStatus" type="gml:MovingObjectStatusType"
substitutionGroup="gml:AbstractTimeSlice"/>

<complexType name="MovingObjectStatusType">
  <complexContent>
    <extension base="gml:AbstractTimeSliceType">
      <sequence>
        <choice>
          <element name="position" type="gml:GeometryPropertyType"/>
          <element ref="gml:pos"/>
          <element ref="gml:locationName"/>
          <element ref="gml:locationReference"/>
          <element ref="gml:location"/>
        </choice>
        <element name="speed" type="gml:MeasureType" minOccurs="0"/>
        <element name="bearing" type="gml:DirectionPropertyType" minOccurs="0"/>
        <element name="acceleration" type="gml:MeasureType" minOccurs="0"/>
        <element name="elevation" type="gml:MeasureType" minOccurs="0"/>
        <element ref="gml:status" minOccurs="0"/>
        <element ref="gml:statusReference" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Елемент *gml:MovingObjectStatus* дозволяє користувачеві описувати поточне місцеположення разом із швидкістю, напрямком, прискоренням та висотою об'єкту у певний часовий зріз.

Додаткову інформацію про поточний статус об'єкту можна записувати в елементах властивості *gml:status* або *gml:statusReference*, задекларованих таким чином:

```
<element name="status" type="gml:StringOrRefType"/>
<element name="statusReference" type="gml:ReferenceType"/>
```

### К.7.3 track

Елемент властивості *gml:track* було скасовано. Натомість треба використовувати *gml:history* (див. пункт 14.5.7). Він задекларований у схемі таким чином:

```
<element name="track" type="gml:HistoryPropertyType" substitutionGroup="gml:history"/>
```

## К.8 Визначення та словники

### К.8.1 DefinitionCollection

```
<element name="DefinitionCollection" type="gml:DictionaryType"
substitutionGroup="gml:Definition"/>
```

Альтернативне ім'я для словників, *gml:DefinitionCollection*, скасовано. Натомість треба використовувати *gml:Dictionary* (див. пункт 15.2.2).

Для посилання на віддалені посилання треба використовувати *gml:dictionaryEntry*; *gml:indirectEntry* скасовано. Якщо об'єкт визначення, що міститься в словнику, використовує властивість *descriptionReference* для посилання на віддалене визначення, то це уможливило включення віддаленого визначення в локальному словнику, надаючи *handle* та ідентифікатор у контексті локального словника.

### К.8.2 definitionMember

```
<element name="definitionMember" type="gml:DictionaryEntryType"
substitutionGroup="gml:dictionaryEntry"/>
```

Альтернативне ім'я *gml:definitionMember* скасовано, натомість треба використовувати *gml:dictionaryEntry* (див. пункт 15.2.3).

### К.8.3 indirectEntry, IndirectEntryType, DefinitionProxy, DefinitionProxyType

Якщо визначення має бути включене через посилання, у його контексті в поточній колекції, то в попередніх версіях GML треба було використовувати скасований елемент властивості *gml:indirectEntry*. Цей елемент задекларований таким чином:

```
<element name="indirectEntry" type="gml:IndirectEntryType"/>
```

```
<complexType name="IndirectEntryType">
  <sequence>
    <element ref="gml:DefinitionProxy"/>
  </sequence>
</complexType>
```

```
<element name="DefinitionProxy" type="gml:DefinitionProxyType" substitutionGroup="gml:Definition"/>
```

```
<complexType name="DefinitionProxyType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
<element ref="gml:definitionRef"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

*gml:DefinitionProxy* надає обов'язкову *handle* (*gml:id*), та містить посилання на визначення, подане десь іще. Очікують, що цей запис буде зручним в уможливленні множинних елементів в одному XML документі, щоб містити скорочені покажчики (XPointer) посилання, що дозволяють зовнішнє визначення, надане в елементі словника у тому ж самому XML документі.

Посилання забезпечено елементом *gml:definitionRef*, задекларованим таким чином:

```
<element name="definitionRef" type="gml:ReferenceType"/>
```

Він використовує *gml:ReferenceType*. Посилання на віддалений запис може бути в словнику в тому ж самому чи іншому XML документі.

## К.9 Одиниці, міри та значення

### К.9.1 dmsAngle

Елемент *gml:dmsAngle* використовували для запису значення кута в форматі “градус-хвилина-секунда” або “градус-хвилина”, але його скасовано включно з усіма залежними компонентами схеми, оскільки треба використовувати десяткову форму запису кута в градусах для читання машинами та обміну між ними (див. пункт 16.3.3). Він використовує таку схему:

```
<element name="dmsAngle" type="gml:DMSAngleType"/>
```

```
<complexType name="DMSAngleType">
```

```
<sequence>
```

```
<element ref="gml:degrees"/>
```

```
<choice minOccurs="0">
```

```
<element ref="gml:decimalMinutes"/>
```

```
<sequence>
```

```
<element ref="gml:minutes"/>
```

```
<element ref="gml:seconds" minOccurs="0"/>
```

```
</sequence>
```

```
</choice>
```

```
</sequence>
```

```
</complexType>
```

```
<element name="degrees" type="gml:DegreesType"/>
```

```
<complexType name="DegreesType">
```

```
<simpleContent>
```

```
<extension base="gml:DegreeValueType">
```

```
<attribute name="direction">
```

```
<simpleType>
```

```
<restriction base="string">
```

```
<enumeration value="N"/>
```

```
<enumeration value="E"/>
```

```
<enumeration value="S"/>
```

```
<enumeration value="W"/>
```

```
<enumeration value="+"/>
```

```
<enumeration value="-"/>
```

```
</restriction>
```

```
</simpleType>
```

```
</attribute>
```

```
</extension>
```

```
</simpleContent>
```

```
</complexType>
```

```
<simpleType name="DegreeValueType">
```

```
<restriction base="nonNegativeInteger">
```

```
<maxInclusive value="359"/>
```

```

</restriction>
</simpleType>

<element name="decimalMinutes" type="gml:DecimalMinutesType"/>

<simpleType name="DecimalMinutesType">
  <restriction base="decimal">
    <minInclusive value="0.00"/>
    <maxExclusive value="60.00"/>
  </restriction>
</simpleType>

<element name="minutes" type="gml:ArcMinutesType"/>

<simpleType name="ArcMinutesType">
  <restriction base="nonNegativeInteger">
    <maxInclusive value="59"/>
  </restriction>
</simpleType>

<element name="seconds" type="gml:ArcSecondsType"/>

<simpleType name="ArcSecondsType">
  <restriction base="decimal">
    <minInclusive value="0.00"/>
    <maxExclusive value="60.00"/>
  </restriction>
</simpleType>

```

### К.9.2 degrees

Елемент *degrees* надає цілочислову величину в градусах з ідентифікацією напрямку кута. Цей елемент призначений для використання в географічних позиціях та має XML атрибут *direction*, що може приймати такі значення:

- "N" або "S" для широти, що означає північ чи південь відносно екватору;
- "E" або "W" для довготи, що означає схід чи захід відносно початкового меридіану;
- "+" або "-" для інших кутів, у визначеному напрямку повороту від визначеного референсного напрямку.

### К.9.3 decimalMinutes

Десяткова кількість арк-хвилин для використання в кутовому значенні в форматі “градус-хвилина”.

### К.9.4 minutes

Цілочислова кількість арк-хвилин для використання в кутовому значенні в форматі “градус-хвилина-секунда”.

### К.9.5 seconds

Кількість арк-секунд для використання в кутовому значенні в форматі градус-хвилина-секунда.

### К.9.6 AngleChoiceType

Для підтримки вибору між різними кодуваннями кутів у моделі контенту надано тип зручності *gml:AngleChoiceType*. Цей елемент містить інший елемент – або *angle* або *dmsAngle*.

Він задекларований у схемі таким чином:

```
<complexType name="AngleChoiceType">
  <choice>
    <element ref="gml:angle"/>
    <element ref="gml:dmsAngle"/>
  </choice>
</complexType>
```

### К.10 Напрямки

Властивості *gml:horizontalAngle* та *gml:verticalAngle* у *gml:DirectionVectorType* (див. підрозділ 17.3) скасовано та замінено на *gml:vector* (див. підпункт 10.1.4.5).

### К.11 Покриття

#### К.11.1 MappingRule

*gml:CovarianceMappingRule* (див. пункт 19.3.12) є заміною скасованого *gml:MappingRule*:

```
<element name="MappingRule" type="gml:StringOrRefType"/>
```

#### К.11.2 IncrementOrder

Скасована властивість *gml:order* має модель контенту (обмежену до двохвимірних покриттів):

```
<simpleType name="IncrementOrder">
  <restriction base="string">
    <enumeration value="+x+y"/>
    <enumeration value="+y+x"/>
    <enumeration value="+x-y"/>
    <enumeration value="-x-y"/>
  </restriction>
</simpleType>
```

Значення переліку у даному випадку вказує на порядок зростання на перших двох осях, тобто "+x-y" означає, що точки на першій вісі треба проходити від найнижчої до найвищої, і точки на другій вісі треба проходити від найвищої до найнижчої. Припускають, що приращення на всіх інших осях (якщо такі є), крім перших двох осей, відбувається від найнижчої точки до найвищої.

Якщо атрибут порядку пропущено, то припускають, що він має значення "+x+y".

Елемент замінено на *gml:axisOrder* (див. пункт 19.3.14).

#### К.11.3 Властивості набору доменів

```
<element name="multiPointDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
<element name="multiCurveDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
<element name="multiSurfaceDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
<element name="multiSolidDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
<element name="gridDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
<element name="rectifiedGridDomain" type="gml:DomainSetType" substitutionGroup="gml:domainSet"/>
```

Ці властивості скасовано, оскільки видалено всі попередні застосування отримання через обмеження в елементах властивості. Див. примітку в пункті 21.2.6. Натомість треба використовувати *gml:domainSet*.

## ДОДАТОК Л (довідковий) МОДУЛЯРИЗАЦІЯ ТА ЗАЛЕЖНОСТІ

GML схема, описана в цьому міжнародному стандарті, модуляризована в додатку В так, щоб допомагати створювати профілі, якщо потрібен тільки тематичний піднабір GML. Наприклад, GML 2.1 прикладна схема, що мігрувала в цей міжнародний стандарт без додавання нових можливостей, могла б визначати профіль, що включає документ схеми *feature.xsd*. Такий профіль не містив би нових визначень 19136 для референцних систем координат, топології, покриттів, динамічних географічних об'єктів та спостережень. Однак, вона містив би всі базові типи, що були додані з часів реалізації GML 2.1.

Модуляризація GML схеми за умовчанням створює залежності між GML базовими схемами, показаними на рисунку Л.1. Пунктирна стрілка на рисунку означає, що схема на початку стрілки залежить від схеми на кінці стрілки. Залежність може виникати там, де одна схема `<include>` (включає) іншу схему в “gml” просторі імен. Наприклад, *feature.xsd* `<include>s geometryBasic2d.xsd` `<import> xlink.xsd` з простору імен “xlink”. Залежність може також виникати там, де одна схема `<import>` (імпортує) іншу схему для простору імен, відмінного від “gml”, наприклад, *gmlBase.xsd* `<import> xlink.xsd` з “xlink” простору імен.

Існує сім документів схеми у GML, від яких не залежить жодний інший документ GML схеми. Ці схеми вищого рівня є коренями ієрархій, які частково перекриваються, документів GML схеми:

- *observation.xsd*;
- *dynamicFeature.xsd*;
- *coverage.xsd*;
- *topology.xsd*;
- *defaultStyle.xsd*;
- *coordinateReferenceSystems.xsd*;
- *temporalReferenceSystems.xsd*.

Прикладна схема, яка потребує визначень з більше ніж однієї з цих ієрархій GML схем тематичних піднаборів, може використовувати документ вищого рівня, розроблений користувачем, що містить множинні `<include>` тільки для відповідних документів схем, тим самим виключаючи небажані визначення GML типу.

Однак, коли прикладну схему будуть використовувати в середовищі обробки, якій бракує потужності процесору, пам'яті або I/O *bandwidth*, наприклад, у мобільному пристрої, то часто потрібний абсолютно мінімальний імпорт GML визначень. Підхід до GML схеми вищого рівня, створеної розробником, описаний вище, може привнести неприйнятно велике число небажаних визначень із кожної GML схеми, включеної в GML профіль документу схеми вищого рівня, розробленого користувачем. Рішенням є створення лише однієї GML схеми піднабору, що містить саме потрібні визначення GML

типу та елементу. Однак, створення такої GML схеми піднабору вручну з використанням тексту чи XML редактора для переміщення та вставки визначень є трудомістким та багатим на помилки процесом, оскільки він передбачає аналіз залежностей визначень типів по багатьох документах GMLсхеми. Натомість рекомендований автоматизований підхід. Інформативний приклад реалізації інструменту GML піднабору схеми включений у додаток Е. Схеми піднабору, хоча й розроблені, є профілями GML, як визначено в розділі 20.

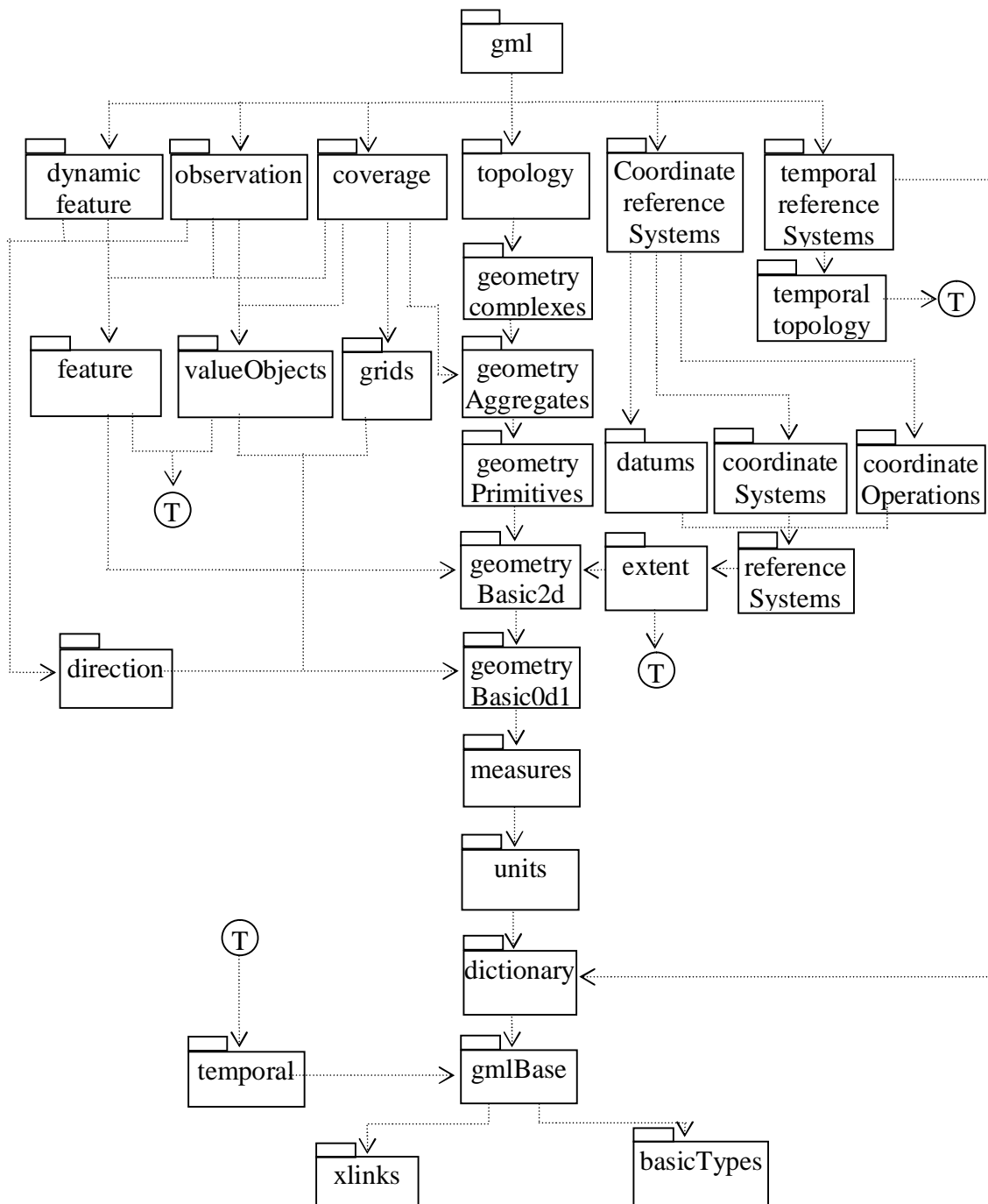


Рисунок Л.1 – Залежності схеми



## БІБЛІОГРАФІЯ

1. ISO 1000, SI units and recommendations for the use of their multiples and of certain other units (ISO 1000 Одиниці системи одиниць та рекомендації використання їх добутоків та певних інших одиниць).
2. ISO 8879, Information processing — Text and office systems — Standard Generalized MarkupLanguage (SGML), Обробка інформації – Текстові та офісні системи – Стандартна узагальнена мова розмітки (SGML).
3. ISO/IEC TR 10000-1:1998 Information technology — Framework and taxonomy of International Standardized Profiles - Part 1: General principles and documentation framework, Інформаційна технологія – Структура та таксономія міжнародних стандартизованих профілів.
4. ISO 19101:2002 Geographic information – Reference Model, Географічна інформація – еталонна модель.
5. ISO 19105:2000, Geographic information — Conformance and testing, Географічна інформація – Відповідність та тестування.
6. ISO 19106:2004, Geographic Information — Profiles, Географічна інформація – профілі.
7. ISO 19110:2005, Geographic Information — Methodology for feature cataloguing, Географічна інформація – Методологія для каталогізації географічних об'єктів.
8. ISO 19117:2005, Geographic Information — Portrayal, Географічна інформація – Зображення
9. ISO 19133, Geographic information — Location-based services — Tracking and navigation, Географічна інформація – Сервіси на базі місцеположення – Відслідковування та навігація.
10. ISO 19137:2007, Geographic information — Core profile of the spatial schema, Географічна інформація – Ядро профілю просторової схеми
11. ISO 19141, Geographic information — Schema for moving features, Географічна інформація – Схема для рухомих географічних об'єктів
12. ISO/IEC 19501:2005, Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2, Інформаційна технологія – Відкритий розподілений процес – Уніфікована мова моделювання (UML), версія 1.4.2.
13. Cover Pages Technology Report at <http://xml.coverpages.org/geographyML.html>
14. Langran, G. Time in Geographic Information Systems. London: Taylor & Francis Ltd. 1992
15. Kaufman, M. & Wagner D. Drawing Graphs, Springer LNCS 2025, 1998.
16. Battista, G., Eades P., Tamassia R. & Tollis I. Graph Drawing, Prentice Hall 1999.
17. OpenGIS® Abstract Specification Topic 5, The OpenGIS Feature, OGC document 99-105r2, Абстрактна специфікація Відкритого геопросторового консорціуму Тема 5, Географічний об'єкт Відкритого геопросторового консорціуму, документ Відкритого геопросторового консорціуму 99-105r2.

18. OpenGIS® Abstract Specification Topic 6, The Coverage Type, OGC document 00-106, Абстрактна специфікація Відкритого геопросторового консорціуму Тема 6, Тип покриття, документ Відкритого геопросторового консорціуму 00-106.
19. OpenGIS® Abstract Specification Topic 8, Relations between Feature, OGC document 99-108r2, Абстрактна специфікація Відкритого геопросторового консорціуму Тема 8, Відношення між географічними об'єктами, документ Відкритого геопросторового консорціуму 99-108r2.
20. OpenGIS® Abstract Specification Topic 10, Feature Collections, OGC document 99-110, Абстрактна специфікація Відкритого геопросторового консорціуму Тема 10, Колекції географічних об'єктів, документ Відкритого геопросторового консорціуму 99-110.
21. OGC Best Practice, Observations and Measurements, OGC document 05-087r4, Передовий досвід відкритого геопросторового консорціуму, Спостереження та вимірювання, документ Відкритого геопросторового консорціуму 05-087r4.
22. UCUM, Unified Code for Units of Measure, Schadow, G. and McDonald, C. J. (eds.), <http://aurora.rg.iupui.edu/UCUM>
23. W3C XML Base, XML Base ,W3C Recommendation (27 June 2001), XML база, W3C рекомендація (27 червня 2001 р.);
24. W3C XPath, W3C XML Path Language (XPath) Version 1.0. W3C Recommendation (16 November, 1999), W3C XML Мова шляхів (XPath) Версія1.0. W3C рекомендація (16 листопада, 1999 р.).