



OGC (POINTS OF INTEREST)

STANDARD

DRAFT

Version: 1.0

Submission Date: 2029-03-30

Approval Date: 2029-03-30

Publication Date: 2023-07-19

Editor: Charles Heazel, Matthew Brian, John Purss

Notice for Drafts: This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: http://portal.opengeospatial.org/public_ogc/change_request.php

Copyright notice

Copyright © 2023 Open Geospatial Consortium
To obtain additional rights of use, visit <http://www.ogc.org/legal/>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I.	ABSTRACT	viii
II.	KEYWORDS	viii
III.	PREFACE	ix
IV.	SECURITY CONSIDERATIONS	x
V.	SUBMITTING ORGANIZATIONS	xi
2.	CONFORMANCE	13
2.1.	Conceptual Models	13
2.2.	Implementation Specifications	13
2.3.	Conformance Classes	14
1.	SCOPE	2
3.	NORMATIVE REFERENCES	4
4.	TERMS AND DEFINITIONS	6
5.	CONVENTIONS	19
5.1.	Identifiers	19
5.2.	UML Notation	19
6.	POI	24
6.1.	Feature Model	24
6.2.	Geometry	26
6.3.	POI ISO Extensions	28
6.4.	POI Class Model	30
6.5.	POI Payload	30
6.6.	POI Data Dictionary	32
7.	MEDIA TYPES FOR ANY DATA ENCODING(S)	43
ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)		45
A.1.	Conformance Class A	45
ANNEX B (INFORMATIVE) ISO DATA DICTIONARY		47
B.1.	General Feature Model	47

B.2. Geometry	49
B.3. Citation and responsible party information	52
B.4. Constraint information	65
B.5. Identification information	69
B.6. Name types	82
B.7. Primitive types	85
 ANNEX C (INFORMATIVE) REVISION HISTORY	 95
 BIBLIOGRAPHY	 97

LIST OF TABLES

Table 1 – Abstract Feature Class	32
Table 2 – Abstract Feature With Lifespan Class	33
Table 3 – Abstract POI Class	35
Table 4 – Feature of Interest Class	39
Table 5 – POI Payload Class	39
Table 6 – Payload Definition Class	40
Table 7 – Payload Schema Class	40
Table 8 – Link Class	41
Table B.1 – Any Feature Class	47
Table B.2 – Attribute Type Class	48
Table B.3 – Feature Type Class	49
Table B.4 – GM_Object Class	50
Table B.5 – GM_Point Class	51
Table B.6 – GM_LineString Class	51
Table B.7 – GM_Polygon Class	52
Table B.8 – CI_Address Class	52
Table B.9 – CI_Citation Class	53
Table B.10 – CI_Contact Class	54
Table B.11 – CI_Date Class	55
Table B.12 – CI_DateTypeCode Codelist	55
Table B.13 – CI_Individual Class	56
Table B.14 – CI_OnLineFunctionCode Codelist	57
Table B.15 – CI_OnlineResource Class	58
Table B.16 – CI_Organisation Class	59
Table B.17 – CI_Party Class	59
Table B.18 – CI_PresentationFormCode Codelist	60
Table B.19 – CI_Responsibility Class	61
Table B.20 – CI_RoleCode Codelist	62

Table B.21 – CI_Series Class	63
Table B.22 – CI_Telephone Class	64
Table B.23 – CI_TelephoneTypeCode Codelist	64
Table B.24 – MD_Classification Class	65
Table B.25 – MD_Constraints Class	66
Table B.26 – MD_LegalConstraints Class	66
Table B.27 – MD_Releasability Class	67
Table B.28 – MD_RestrictionCode Codelist	68
Table B.29 – MD_SecurityConstraints Class	69
Table B.30 – DS_AssociationTypeCode Codelist	70
Table B.31 – DS_InitiativeTypeCode Codelist	70
Table B.32 – MD_AssociatedResource Class	71
Table B.33 – MD_Dataldentity Class	72
Table B.34 – MD_Identification Class	73
Table B.35 – MD_KeywordClass Class	74
Table B.36 – MD_Keywords Class	75
Table B.37 – MD_KeywordTypeCode Codelist	76
Table B.38 – MD_ProgressCode Codelist	77
Table B.39 – MD_REpresentativeFraction Class	78
Table B.40 – MD_Resolution Class	78
Table B.41 – MD_SpatialRepresentationTypeCode Codelist	79
Table B.42 – MD_Usage Class	80
Table B.43 – MD_TopicCategoryCode Codelist	80
Table B.44 – Generic Name Class	83
Table B.45 – Local Name Class	83
Table B.46 – Member Name Class	83
Table B.47 – Namespace Class	84
Table B.48 – Scoped Name Class	84
Table B.49 – Type Name Class	85
Table B.50 – Date Class	85
Table B.51 – DateTime Class	86
Table B.52 – Time Class	86
Table B.53 – Decimal Class	87
Table B.54 – Integer Class	87
Table B.55 – Number Class	87
Table B.56 – Real Class	88
Table B.57 – Unlimited Integer Class	88
Table B.58 – Vector Class	88
Table B.59 – Character Set Code Codelist	89
Table B.60 – Character Class	89
Table B.61 – Character String Class	90

Table B.62 – URI Class	90
Table B.63 – Continuous Truth Class	91
Table B.64 – Discrete Truth Class	91
Table B.65 – Probability Class	91
Table B.66 – Truth Class	92
Table B.67 – Boolean Enumeration	92
Table B.68 – Logical Enumeration	93
Table C.1	95

LIST OF FIGURES

Figure 1	19
Figure 2 – UML notation (see ISO TS 19103, Geographic information - Conceptual schema language).	20
Figure 3 – Example UML diagram demonstrating the UML notation and coloring scheme used throughout the POI Standard.	22
Figure 4 – Feature Model	25
Figure 5 – Geometry Model	27
Figure 6 – POI UML Model - ISO Extensions	29
Figure 7 – POI UML Model - Core	30
Figure 8 – POI UML Model - Payload	31

LIST OF RECOMMENDATIONS

REQUIREMENT 1: REQUIREMENT – GENERAL FEATURE MODEL: /req/core/req-generalfeaturemodel	26
REQUIREMENT 2: REQUIREMENT – GEOMETRY: /req/core/req-geometry	28
REQUIREMENT 3: REQUIREMENT – ABSTRACT FEATURE: /req/core/req-abstractfeature	32
REQUIREMENT 4: REQUIREMENT – ABSTRACT FEATURE DESCRIPTION: /req/core/req-abstractfeature-description	33
REQUIREMENT 5: REQUIREMENT – ABSTRACT FEATURE FEATURE ID: /req/core/req-abstractfeature-featureid	33
REQUIREMENT 6: REQUIREMENT – ABSTRACT FEATURE IDENTIFIER: /req/core/req-abstractfeature-identifier	33
REQUIREMENT 7: REQUIREMENT – ABSTRACT FEATURE NAME: /req/core/req-abstractfeature-name	33

REQUIREMENT 8: REQUIREMENT – FEATURE WITH LIFESPAN: /req/core/req-featurewithlifespan	34
REQUIREMENT 9: REQUIREMENT – FEATURE WITH LIFESPAN CREATION DATE: /req/core/req-featurewithlifespan-creationdate	34
REQUIREMENT 10: REQUIREMENT – FEATURE WITH LIFESPAN TERMINATION DATE: /req/core/req-featurewithlifespan-terminationdate	35
REQUIREMENT 11: REQUIREMENT – FEATURE WITH LIFESPAN VALID FROM: /req/core/req-featurewithlifespan-validfrom	35
REQUIREMENT 12: REQUIREMENT – FEATURE WITH LIFESPAN VALID TO: /req/core/req-featurewithlifespan-validto	35
REQUIREMENT 13: REQUIREMENT – POI CLASS: /req/core/req-poi-class	36
REQUIREMENT 14: REQUIREMENT – POI CONTACT INFORMATION: /req/core/req-poi-contactInfo	37
REQUIREMENT 15: REQUIREMENT – POI DEFINITION: /req/core/req-poi-definition	37
REQUIREMENT 16: REQUIREMENT – POI FEATURE OF INTEREST: /req/core/req-poi-featureOfInterest	37
REQUIREMENT 17: REQUIREMENT – POI METADATA: /req/core/req-poi-metadata	38
REQUIREMENT 18: REQUIREMENT – POI PAYLOAD: /req/core/req-poi-payload	38
REQUIREMENT 19: REQUIREMENT – POI KEYWORDS: /req/core/req-poi-keywords	38
REQUIREMENT 20: REQUIREMENT – POI RIGHTS: /req/core/req-poi-rights	38
REQUIREMENT 21: REQUIREMENT – POI SCHEMA: /req/core/req-poi-schema	38
REQUIREMENT 22: REQUIREMENT – POI SYMBOLOGY: /req/core/req-poi-symbology	39
REQUIREMENT 23: REQUIREMENT FEATURE OF INTEREST: /req/core/req-feature-of-interest	39
REQUIREMENT 24: REQUIREMENT – POI-PAYLOAD: /req/core/req-poi_payload-class	40
REQUIREMENT A.1: /req/req-class-a/req-name-1	45

ABSTRACT

<Insert Abstract Text here>

KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, API, openapi, html

PREFACE

NOTE: Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

SECURITY CONSIDERATIONS

No security considerations have been made for this document.

SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Organization One
- Organization Two

2

CONFORMANCE

CONFORMANCE

This standard defines a Conceptual Model which is independent of any encoding or formatting techniques. The Standardization Targets for this standard are:

1. Conceptual Models (extended versions of this conceptual model)
2. Implementation Specifications (encodings of this conceptual model)

2.1. Conceptual Models

A Conceptual Model standardization target is a version of the POI Conceptual Model (CM) tailored for a specific user community. This tailoring can include:

1. Reduction of the multiplicity for an attribute or association
2. Restriction on the valid values for an attribute
3. Additional POI properties documented through the POIPayload class.

Of these options, actions #1 and #2 can be performed when creating an implementation specification. Only action #3 requires an extension of the POI conceptual model. These extensions are accomplished using the POI Payload mechanism described in POI Payload.

2.2. Implementation Specifications

Implementation Specifications define how a Conceptual Model should be implemented using a specific technology. Conformant Implementation Specifications provide evidence that they are an accurate representation of the Conceptual Model. This evidence should include implementations of the abstract tests specified in Annex A (normative) of this document.

Since this standard is agnostic to the implementing technologies, the specific techniques to be used for conformance testing cannot be specified. Implementation Specifications need to provide evidence of conformance which is appropriate for the implementing technologies. This evidence should be provided as an annex to the Implementation Specification document.

2.3. Conformance Classes

This standard identifies one “Core” conformance class. This conformance class defines the conformance criteria for the requirements defined in one “Core” requirements class. The tests for each conformance class are documented in Annex A. These tests are organized by Requirements Class. So an implementation of the Core conformance class must pass all tests specified in Annex A for the Core requirements class.

The POI Conceptual Model is defined by the POI UML model. This standard is a representation of that UML model in document form. In the case of a discrepancy between the UML model and this document, the UML model takes precedence.

1

SCOPE

SCOPE

This document describes a data model for representing information about points of interest (POI).

In the most broad terms, a “point of interest” is a location about which information of general interest is available. A POI can be as simple as a set of coordinates and an identifier, or more complex such as a three dimensional model of a building with names in various languages, information about open and closed hours, and a civic address.

POI data has many uses including navigation systems, mapping, geocaching, location-based social networking games, and augmented reality browsers.

POI data has traditionally been exchanged in proprietary formats by various transport mechanisms. This specification defines a flexible, lightweight, extensible POI data model. This will enable content publishers to effectively describe and efficiently serve and exchange POI data.

To achieve these goals, this document describes a generic data model that may be instantiated in a variety of serializations, including XML, JSON and RDF. The data model is designed to be extended with POI information specific to the geospatial data it represents.

3

NORMATIVE REFERENCES

NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

at www.opengis.net

at [Object Management Group \(OMG\) UML Standard Version 2.5.1](http://Object Management Group (OMG) UML Standard Version 2.5.1)

IETF: IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*. (January 2005)

ISO: ISO 19101-1:2014, *Geographic information – Reference model – Part 1: Fundamentals*

ISO: ISO 19103:2015, *Geographic Information – Conceptual Schema Language*

ISO: ISO 19107:2003, *Geographic Information – Spatial Schema*

ISO: ISO 19108:2002/Cor 1:2006, *Geographic information – Temporal schema – Technical Corrigendum 1*

ISO: ISO 19109:2015, *Geographic Information – Rules for Application Schemas*

ISO: ISO 19111:2019, *Geographic information – Referencing by coordinates*

ISO: ISO 19115-1:2014, *Geographic Information – Metadata – Part 1: Fundamentals*

ISO/IEC 19507:2012, *Information technology – Object Management Group Object Constraint Language (OCL)*

4

TERMS AND DEFINITIONS

TERMS AND DEFINITIONS

This document uses the terms defined in [OGC Policy Directive 49](#), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications ([OGC 08-131r3](#)), also known as the 'ModSpec'. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

This document uses the terms defined in [OGC Policy Directive 49](#), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this Standard and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications ([OGC 08-131r3](#)), also known as the 'ModSpec'. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

4.1. application schema::

conceptual schema (Clause 4.8) for data required by one or more applications.

Note 1 to entry: An *application schema* (Clause 4.1) contains selected parts of the base schemas presented in the ORM Information Viewpoint.

Note 2 to entry: Designers of *application schemas* (Clause 4.1) may extend or restrict the types defined in the base schemas to define appropriate types for an application *domain* (Clause 4.15).

Note 3 to entry: *Application schemas* (Clause 4.1) are information models for a specific information community.

[**SOURCE:** ISO 19101-1, Clause 4.1.2, modified – OGC Definitions Register at <http://www.opengis.net/def/glossary/term/ApplicationSchema>]

4.2. attribute::

named *property* (Clause 4.41) of an entity

Note 1 to entry: Describes a geometrical, topological, thematic, or other characteristic of an entity.

[SOURCE: ISO/IEC 2382, Clause 2121440]

4.3. class::

description of a set of *objects* (Clause 4.31) that share the same *attributes* (Clause 4.2), *operations* (Clause 4.33), methods, relationships, and semantics

Note 1 to entry: A *class* (Clause 4.3) may use a set of interfaces to specify collections of *operations* (Clause 4.33) it provides to its environment. The term was first used in this way in the general theory of object-oriented programming, and later adopted for use in this same sense in UML.

[SOURCE: ISO 19103, Clause 4.27, modified – Note 1 to entry has been added from ISO 19117:2012, 4.2]

4.4. code::

representation of a label according to a specified scheme

[SOURCE: ISO 19118, Clause 4.3]

4.5. codelist::

value *domain* (Clause 4.15) including a *code* (Clause 4.4) for each permissible value.

[SOURCE: ISO 19136-1, Clause 3.1.7]

4.6. concept::

unit of knowledge created by a unique combination of characteristics

Note 1 to entry: Concepts (Clause 4.6) are not necessarily bound to particular languages. They are, however, influenced by the social or cultural background which often leads to different categorizations.

[SOURCE: ISO 1087-1, Clause 3.2.1]

4.7. conceptual model::

model that defines concepts (Clause 4.6) of a universe of discourse

[SOURCE: ISO 19101-1, Clause 4.1.5]

4.8. conceptual schema::

1. formal description of a *conceptual model* (Clause 4.7) [ISO 19101-1:2014, 4.1.6]
2. base schema. Formal description of the model of any geospatial information. *Application schemas* (Clause 4.1) are built from *conceptual schemas* (Clause 4.8).

[SOURCE: OGC Definitions Register]

4.9. coordinate::

one of a sequence of numbers designating the *position* (Clause 4.40) of a *point* (Clause 4.38)

Note 1 to entry: In a spatial *coordinate reference system* (Clause 4.10), the *coordinate* (Clause 4.9) numbers are qualified by units.

[SOURCE: ISO 19111, Clause 3.1.5]

4.10. coordinate reference system::

coordinate system that is related to an *object* (Clause 4.31) by a *datum* (Clause 4.13)

Note 1 to entry: For geodetic and vertical *datums* (Clause 4.13), the *object* (Clause 4.31) will be the Earth.

[SOURCE: ISO 19111, Clause 3.1.9]

4.11. coordinate system::

set of mathematical rules for specifying how *coordinates* (Clause 4.9) are to be assigned to *points* (Clause 4.38)

[SOURCE: ISO 19111, Clause 3.1.11]

4.12. data type::

specification of a value *domain* (Clause 4.15) with *operations* (Clause 4.33) allowed on values in this *domain* (Clause 4.15)

Note 1 to entry: Data types include primitive predefined types and user-definable types.

Example Integer, Real, Boolean, String, Date and SG Point (conversion of data into a series of *codes* (Clause 4.4)).

[SOURCE: ISO 19103, Clause 4.14]

4.13. dataum::

parameter or set of parameters that realize the *position* (Clause 4.40) of the origin, the scale, and the orientation of a *coordinate system* (Clause 4.11)

[SOURCE: ISO 19111, Clause 3.1.15]

4.14. direct position::

position (Clause 4.40) described by a single set of *coordinates* (Clause 4.9) within a *coordinate reference system* (Clause 4.10)

[SOURCE: ISO 19136-1, Clause 3.1.20]

4.15. domain::

well-defined set

Note 1 to entry: *Domains* (Clause 4.15) are used to define the *domain* (Clause 4.15) set and range set of *attributes* (Clause 4.2), operators and functions.

[SOURCE: ISO 19109, Clause 4.8]

4.16. domain <general vocabulary>::

distinct area of human knowledge to which a terminological entry is assigned

Note 1 to entry: Within a database or other terminology collection, a set of *domains* (Clause 4.15) will generally be defined. More than one *domain* (Clause 4.15) can be associated with a given *concept* (Clause 4.6).

[SOURCE: ISO 19104, Clause 4.11]

4.17. domain <ontology>::

restriction to constrain the subject *class* (Clause 4.3) which participates in a subject-predicate-object triple

[SOURCE: ISO 19150-4, Clause 3.1.12]

4.18. domain <postal address>::

an area in which a set of specific postal address types and postal address renderings is prescribed by postal operators

Example The most typical example of a postal address *domain* (Clause 4.15) is a country where a designated postal operator provides postal delivery services.

[SOURCE: ISO 19160-4, Clause 3.14]

4.19. feature::

abstraction of real-world phenomena

Note 1 to entry: A *feature* (Clause 4.19) may occur as a type or an instance. In this document, *feature* (Clause 4.19) instance is meant unless otherwise specified.

[SOURCE: ISO 19101-1, Clause 4.1.11, modified – Note 1 to entry has been added from ISO 19156, 4.6]

4.20. feature type::

class (Clause 4.3) of *features* (Clause 4.19) having common characteristics

[SOURCE: ISO 19156, Clause 4.7]

4.21. geometric aggregate::

collection of *geometric objects* (Clause 4.25) that has no internal structure

Note 1 to entry: No assumptions about the spatial relationships between the elements can be made.

[SOURCE: ISO 19107, Clause 3.45]

4.22. geometric boundary::

boundary represented by a set of *geometric primitives* (Clause 4.26) that limits the extent of a *geometric object* (Clause 4.25)

[SOURCE: ISO 19107, Clause 3.46]

4.23. geometric complex::

set of disjoint *geometric primitives* (Clause 4.26) where the boundary of each *geometric primitive* (Clause 4.26) can be represented as the union of other *geometric primitives* (Clause 4.26) of smaller dimensions within the same set

Note 1 to entry: The *geometric primitives* (Clause 4.26) in the set are disjoint in the sense that no *direct position* (Clause 4.14) is interior to more than one *geometric primitive* (Clause 4.26). The set is closed under boundary *operations* (Clause 4.33), meaning that for each element in the *geometric complex* (Clause 4.23), there is a collection (also a *geometric complex* (Clause 4.23)) of *geometric primitives* (Clause 4.26) that represents the boundary of that element. Recall that the boundary of a *point* (Clause 4.38) (the only 0D primitive *object* (Clause 4.31) type in geometry) is empty. Thus, if the largest dimension *geometric primitive* (Clause 4.26) is a solid (3D), the composition of the boundary operator in this definition terminates after at most three steps. It is also the case that the boundary of any *object* (Clause 4.31) is a cycle.

[SOURCE: ISO 19107, Clause 3.47]

4.24. geometric dimension::

largest number n such that each *point* (Clause 4.38) in a set of *points* (Clause 4.38) can be associated with a subset that has that point in its interior and is topologically isomorphic to \mathbb{E}_n , Euclidean n -space

[SOURCE: ISO 19107, Clause 3.48]

4.25. geometric object::

spatial object (Clause 4.31) representing a *geometric set* (Clause 4.27)

Note 1 to entry: A *geometric object* (Clause 4.25) consists of a *geometric primitive* (Clause 4.26), a collection of *geometric primitives* (Clause 4.26), or a *geometric complex* (Clause 4.23) treated as a single entity. A *geometric object* (Clause 4.25) may be the spatial representation of an *object* (Clause 4.31) such as a *feature* (Clause 4.19) or a significant part of a *feature* (Clause 4.19).

[SOURCE: ISO 19107, Clause 3.49]

4.26. geometric primitive (geometry)::

geometric object (Clause 4.25) representing a single, connected, homogeneous (isotropic) element of space

Note 1 to entry: *Geometric primitives* (Clause 4.26) are non-decomposed *objects* (Clause 4.31) that present information about geometric configuration. They include *points* (Clause 4.38), curves, surfaces, and solids. Many *geometric objects* (Clause 4.25) behave like primitives (supporting the same interfaces defined for geometric primitives) but are actually composites composed of some number of other primitives. General collections may be aggregates and incapable of acting like a primitive (such as the lines of a complex network, which is not connected and thus incapable of being traceable as a single line). By this definition, a *geometric primitive* (Clause 4.26) is topological open, since the boundary *points* (Clause 4.38) are not isotropic to the interior *points* (Clause 4.38). Geometry is assumed to be closed. For *points* (Clause 4.38), the boundary is empty.

[SOURCE: ISO 19107, Clause 3.50]

4.27. geometric set::

set of *points* (Clause 4.38)

[SOURCE: ISO 19107, Clause 3.53]

4.28. location::

particular *place* (Clause 4.34) or *position* (Clause 4.40)

Note 1 to entry: A *location* (Clause 4.28) identifies a geographic *place*.

Note 2 to entry: *Locations* (Clause 4.28) are physically fixed *points* (Clause 4.38), typically on the surface of the Earth, although *locations* (Clause 4.28) can be relative to other, non-earth centric coordinate reference systems.

Note 3 to entry: *Locations* (Clause 4.28) can be a single *point* (Clause 4.38), a centroid, a minimum bounding rectangle, or a set of vectors.

Note 4 to entry: A *location* (Clause 4.28) should be persistent over time and does not change.

Note 5 to entry: Multiple *POIs* (Clause 4.39) may share the same *location* (Clause 4.28).

Note 6 to entry: When a *POI* (Clause 4.39) physically moves it is understood to have acquired a new *location* (Clause 4.28).

[SOURCE: ISO 19112, Clause 3.1.3]

4.29. metaclass::

a *class* (Clause 4.3) whose instances are also *classes* (Clause 4.3)

4.30. method::

implementation of an *operation* (Clause 4.33)

Note 1 to entry: It specifies the algorithm or procedure associated with an *operation* (Clause 4.33).

[SOURCE: ISO/IEC 19501]

4.31. object::

entity with a well defined boundary and identity that encapsulates state and behaviour

Note 1 to entry: This term was first used in this way in the general theory of object oriented programming, and later adopted for use in this same sense in UML. An *object* (Clause 4.31)

is an instance of a *class* (Clause 4.3). *Attributes* (Clause 4.2) and relationships represent state. *Operations* (Clause 4.33), methods, and state machines represent behavior.

[**SOURCE:** version 1.3, 1997.]

4.32. OGC implementation specification::

OGC implementation specification (Clause 4.32) document type defined on the OGC Document Types Register

4.33. operation::

specification of a transformation or query that an *object* (Clause 4.31) may be called to execute

Note 1 to entry: An *operation* (Clause 4.33) has a name and a list of parameters.

Note 2 to entry: See ISO 19119, Clause 7.2 for a discussion of *operation* (Clause 4.33)

[**SOURCE:** ISO 19119, Clause 4.1.10]

4.34. place::

identifiable part of any space

[**SOURCE:** ISO 19155, Clause 4.8]

4.35. Platform (Model Driven Architecture)::

the set of resources on which a system is realized.

[**SOURCE:** OMG Model Driven Architecture Guide, modified – [Object Management Group, Model Driven Architecture Guide rev. 2.0](#)]

4.36. Platform Independent Model:

a model that is independent of a specific *platform* (Clause 4.35)

[**SOURCE:** OMG Model Driven Architecture Guide, modified – [Object Management Group, Model Driven Architecture Guide rev. 2.0](#)]

4.37. Platform Specific Model:

a model of a system that is defined in terms of a specific *platform* (Clause 4.35)

[**SOURCE:** OMG Model Driven Architecture Guide, modified – [Object Management Group, Model Driven Architecture Guide rev. 2.0](#)]

4.38. point::

0-dimensional geometric primitive, representing a *position* (Clause 4.40)

[**SOURCE:** ISO 19136-1, Clause 3.1.47]

4.39. point of interest::

POI ADMITTED

location (Clause 4.28) where one can find a *place*, product or service

Note 1 to entry: A POI (Clause 4.39) is typically identified by *name* rather than by an *address*.

Note 2 to entry: A POI (Clause 4.39) is characterized by *type*, which may be used as a reference *point* (Clause 4.38) or a target in a *location* (Clause 4.28) based service request.

Note 3 to entry: A POI (Clause 4.39) does not exclude the labeling, identification, and tracking of persons and other physical *objects* (Clause 4.31) that have no permanent *location* (Clause 4.28).

Example destination of a route; such as, Boston

4.40. position::

data type that describes a *point* (Clause 4.38) or *geometry* potentially occupied by an *object* (Clause 4.31) or *person*

Note 1 to entry: A *direct position* (Clause 4.14) is a semantic subtype of *position* (Clause 4.40). *Direct positions* (Clause 4.14) as described can only define a *point* (Clause 4.38), and therefore not all *positions* (Clause 4.40) can be represented by a *direct position* (Clause 4.14). That is consistent with the *is* type of relation. An ISO 19107 geometry is also a *position* (Clause 4.40), but not a *direct position* (Clause 4.14)

[SOURCE: ISO 19133]

4.41. property::

facet or *attribute* (Clause 4.2) of an *object* (Clause 4.31) referenced by a name

Example Abby's car has the colour red, where "colour red" is a *property* (Clause 4.41) of the car.

[SOURCE: ISO 19143, Clause 4.21, modified – Note 1 to entry has been added from ISO 19156, 4.15]

4.42. stereotype::

extension of an existing *metaclass* (Clause 4.29) that enables the use of *platform* (Clause 4.35) or *domain* (Clause 4.15) specific terminology or notation in place of, or in addition to, the ones used for the extended *metaclass* (Clause 4.29)

[SOURCE: ISO 19150-2, Clause 4.1.35]

5

CONVENTIONS

5.1. Identifiers

The normative provisions in this document are denoted by the URI

<http://www.opengis.net/spec/POI/1.0>

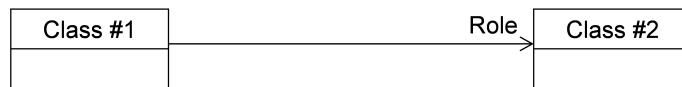
Figure 1

All requirements and conformance tests that appear in this document are denoted by partial URIs relative to this base.

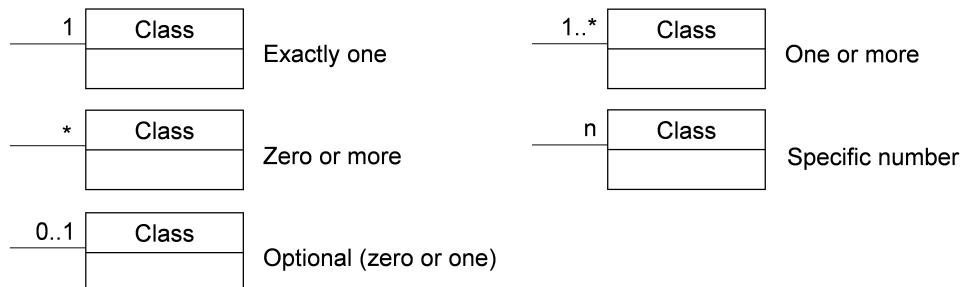
5.2. UML Notation

The POI Conceptual Model (CM) Standard is presented in this document through diagrams using the Unified Modeling Language (UML) static structure diagram (see Booch et al. 1997). The UML notations used in this standard are described in the diagram in Figure 2.

Association between classes



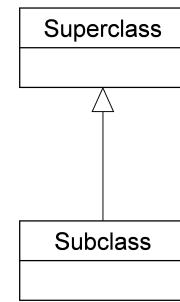
Association cardinality



Aggregation between classes



Class inheritance



Composition between classes



Figure 2 – UML notation (see ISO TS 19103, Geographic information - Conceptual schema language).

All associations between model elements in the POI Conceptual Model are uni-directional. Thus, associations in the model are navigable in only one direction. The direction of navigation is depicted by an arrowhead. In general, the context an element takes within the association is indicated by its role. The role is displayed near the target of the association. If the graphical representation is ambiguous though, the position of the role has to be drawn to the element the association points to.

The following stereotypes are used in this model:

- «FeatureType» represents features that are similar and exhibit common characteristics. Features are abstractions of real-world phenomena and have an identity.
- «TopLevelFeatureType» denotes features that represent the main components of the conceptual model. Top-level features may be further semantically and spatially decomposed and substructured into parts.
- «Type» denotes classes that are not directly instantiable, but are used as an abstract collection of operation, attribute and relation signatures. The stereotype is used in the

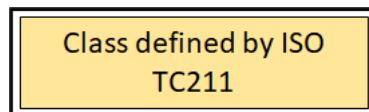
POI Conceptual Model only for classes that are imported from the ISO standards 19103, 19107, 19109, and 19115.

- «ObjectType» represents objects that have an identity, but are not features.
- «DataType» defines a set of properties that lack identity. A data type is a classifier with no operations, whose primary purpose is to hold information.
- «Enumeration» enumerates the valid attribute values in a fixed list of named literal values. Enumerations are specified in the POI Conceptual Model.
- «BasicType» defines a basic data type.
- «CodeList» enumerates the valid attribute values. In contrast to Enumeration, the list of values is open and, thus, not given inline in the POI UML Model. The allowed values can be provided within an external code list.
- «Union» is a list of attributes. The semantics are that only one of the attributes can be present at any time.
- «Property» denotes attributes and association roles. This stereotype does not add further semantics to the conceptual model, but is required to be able to add tagged values to the attributes and association roles that are relevant for the encoding.

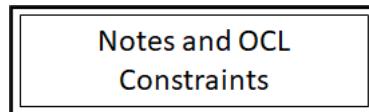
In order to enhance the readability of the POI UML diagrams, classes are depicted in different colors. The following coloring scheme is applied:



Classes painted in green belong to the POI Requirements Class.



Classes painted in yellow are defined in the ISO standards 19107, 19109, or 19115. Their class names are preceded by the UML package name, in which the classes are defined.



The color white is used for notes and Object Constraint Language (OCL) constraints that are provided in the UML diagrams.

The example UML diagram in Figure 3 demonstrates the UML notation and coloring scheme used throughout this standard. In this example, the yellow classes are associated with the

CityGML Building module, the blue classes are from the CityGML Core and Construction modules, and the green class depicts a geometry element defined by ISO 19107.

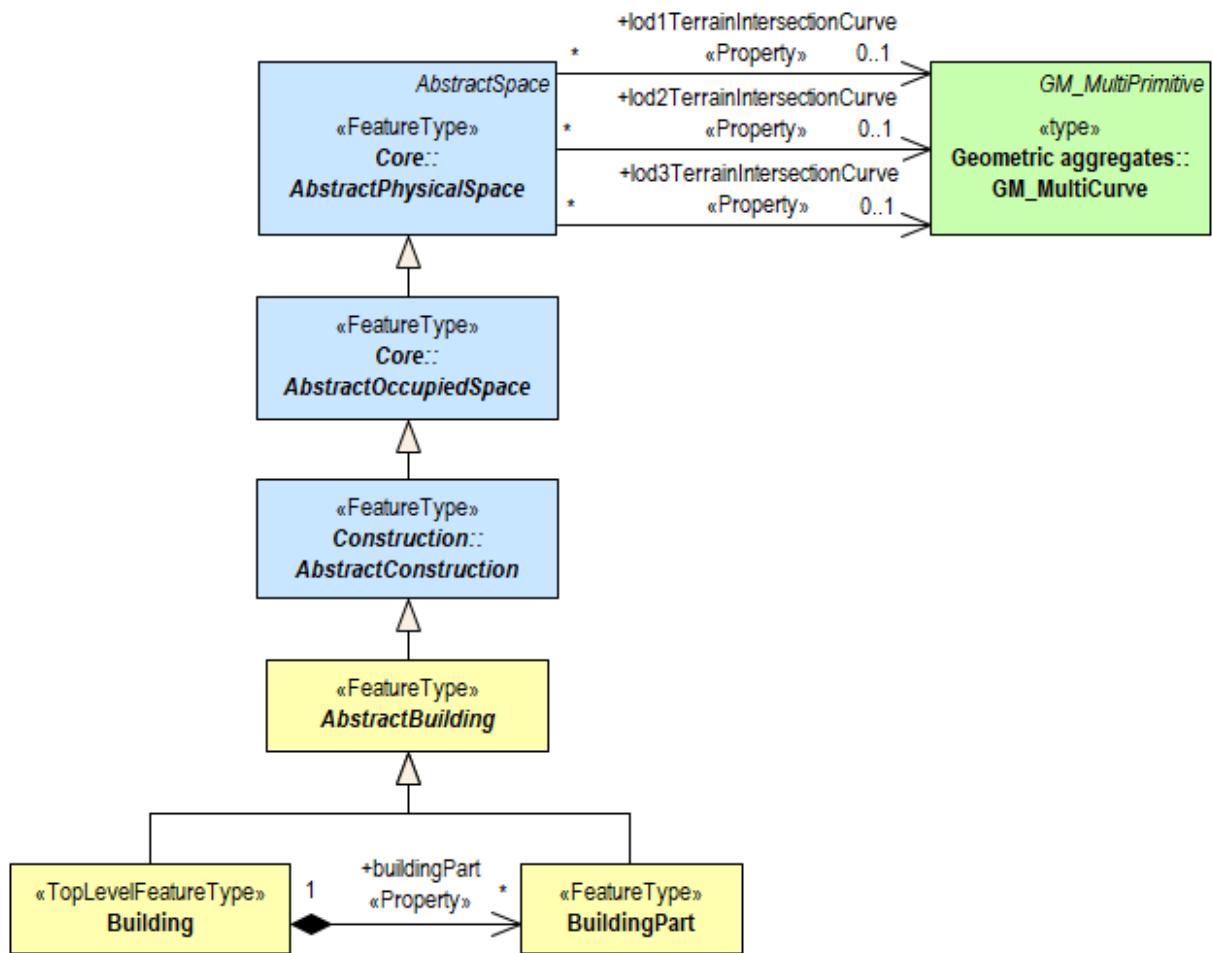


Figure 3 – Example UML diagram demonstrating the UML notation and coloring scheme used throughout the POI Standard.

6

POI

6.1. Feature Model

A Point of Interest (POI) is a Feature. Therefore, it is important to understand what a POI inherits from the OGC Feature model.

The OGC Feature Model is defined in ISO 19109:2015 Geographic Information – Rules for application schema. A UML model showing applicable portions of the General Feature Model is provided in Figure 1.

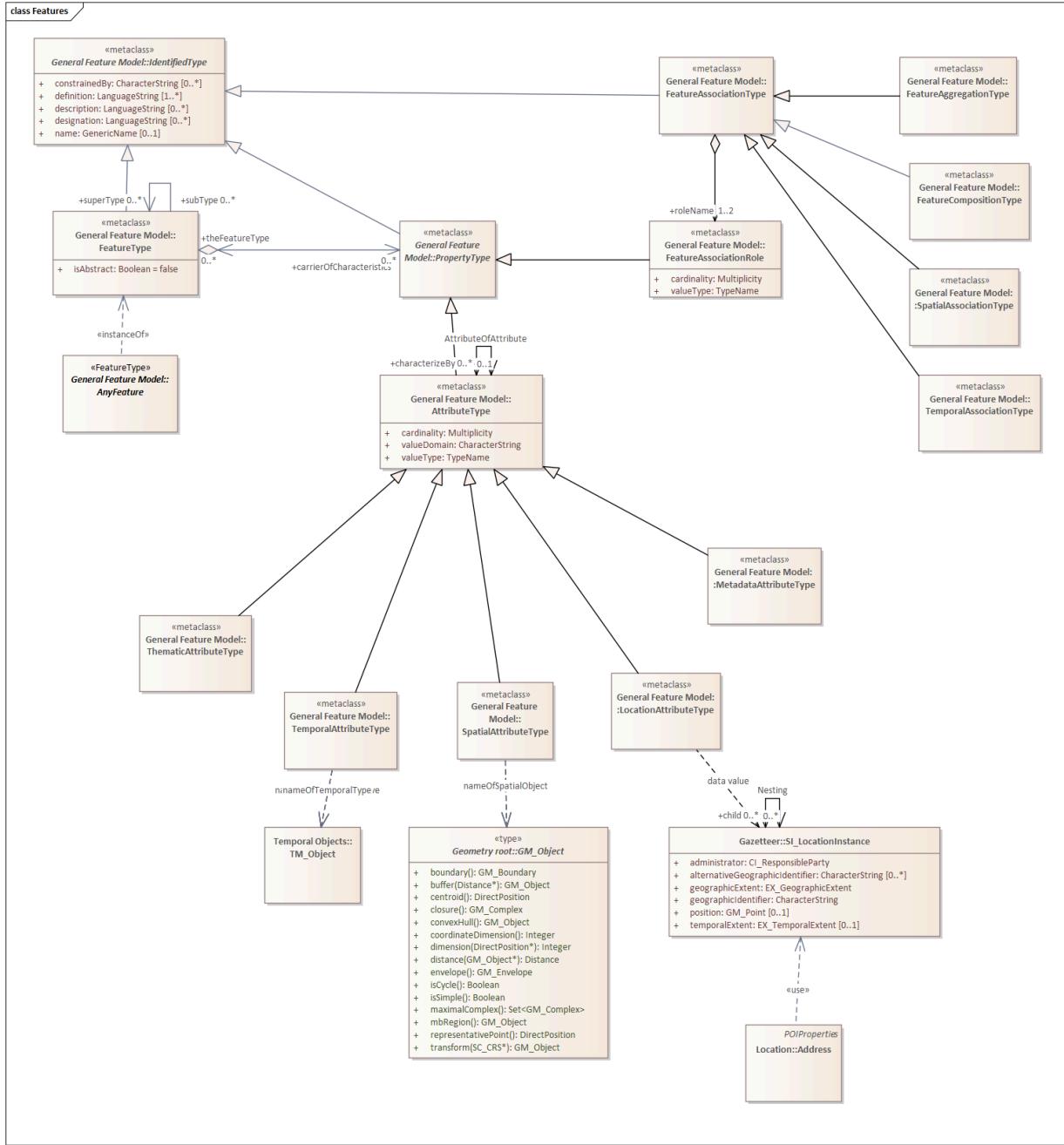


Figure 4 – Feature Model

The most relevant classes defined by this model are described below:

FeatureType: This class describes how a feature class shall be constructed in an Application Schema. In accordance with the conformance clause of the standard, instances of this class are instantiated as feature classes in an Application Schema

AnyFeature: The class AnyFeature is an instance of the «metaclass» FeatureType (ISO 19109). It represents the set of all classes which are feature types.

In an implementation this abstract class shall be substituted by a concrete class representing a feature type from an application schema associated with a domain of discourse (ISO 19109, ISO 19101).

AttributeType: A characteristic of a feature. The Feature Geometry is one type of attribute.

REQUIREMENT 1: REQUIREMENT – GENERAL FEATURE MODEL

/req/core/req-generalfeaturemodel

-
- | | |
|----------|---|
| A | A encoding of the POI Conceptual Model SHALL be compliant with the General Feature Model defined in ISO 19109. |
| B | A POI instance SHALL include a spatial geometry property using the SpatialAttributeType attribute type and GM_Object class. |
| C | The spatial geometry properties of all POI instances SHALL be defined using the GM_Object class. |
-

6.2. Geometry

The OGC Geometry model is defined in ISO 19107:2003 – Geographic Information – Spatial schema. While there is a new version of this standard, it has not been widely implemented. So the 2003 version has been used in this Standard.

The OGC Geometry Model is capable of representing very complex geometries. Much more complex than are needed for a POI. Therefore, POI geometries are restricted to Points, lines, and Polygons. Figure 2 provides a UML model of the classes from ISO 19107 which are applicable to POIs.

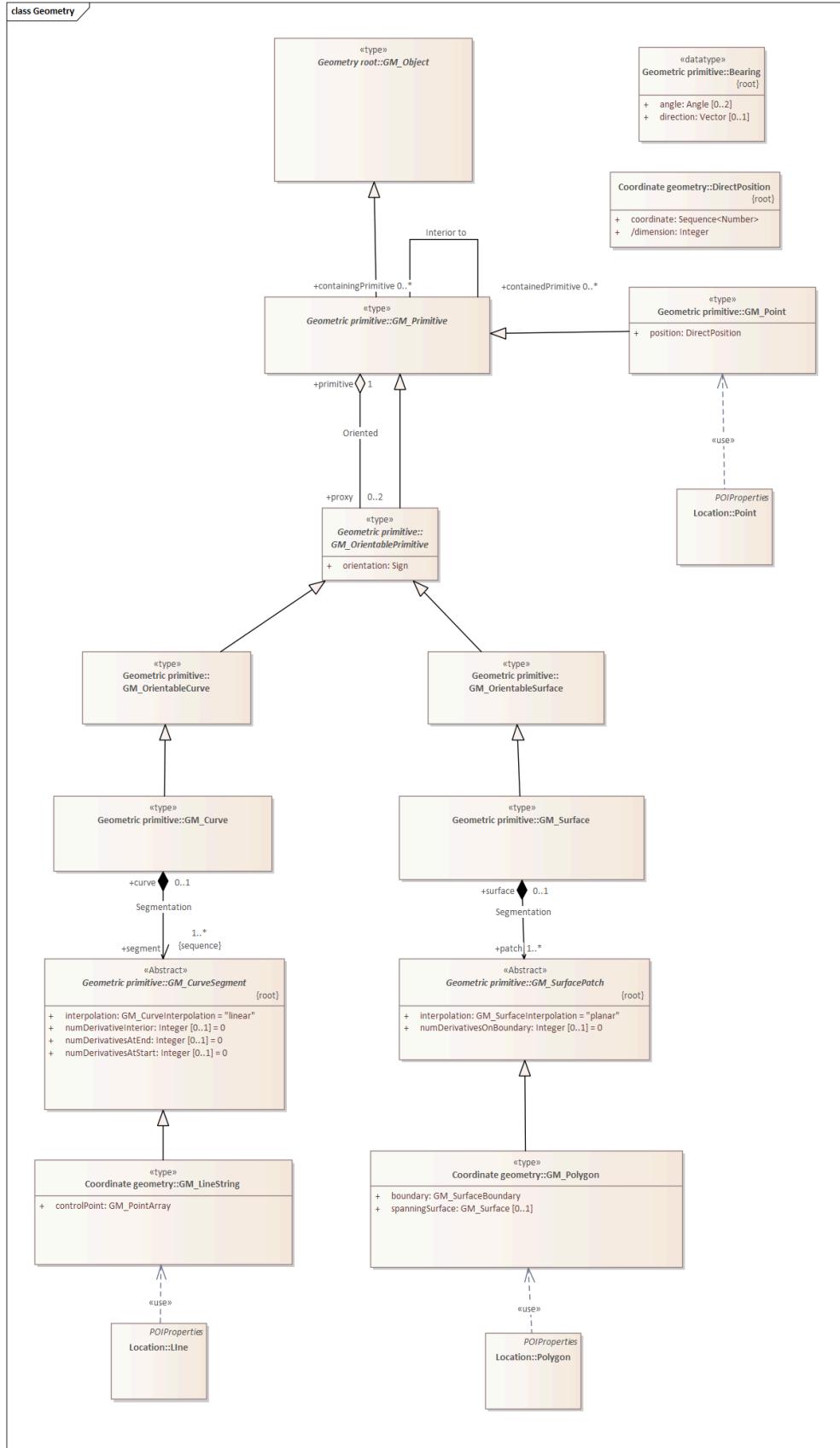


Figure 5 – Geometry Model

The key classes described in this figure are:

GM_Object: Root class for all OGC geometries.

GM_Point: The geometric primitive for Points

GM_LineString: The geometric primitive for line strings.

GM_Polygon: The geometric primitive for areas.

REQUIREMENT 2: REQUIREMENT – GEOMETRY

/req/core/req-geometry

A	The POI Conceptual Model spatial geometry properties SHALL be compliant with the Geometry Model defined in ISO 19107.
B	<p>POI Spatial geometry properties SHALL be one or more of the following:</p> <ul style="list-style-type: none">• GM_Point• GM_LineString• GM_Polygon

6.3. POI ISO Extensions

In this Standard we extend the General Feature Model to support the concept of a Point of Interest.

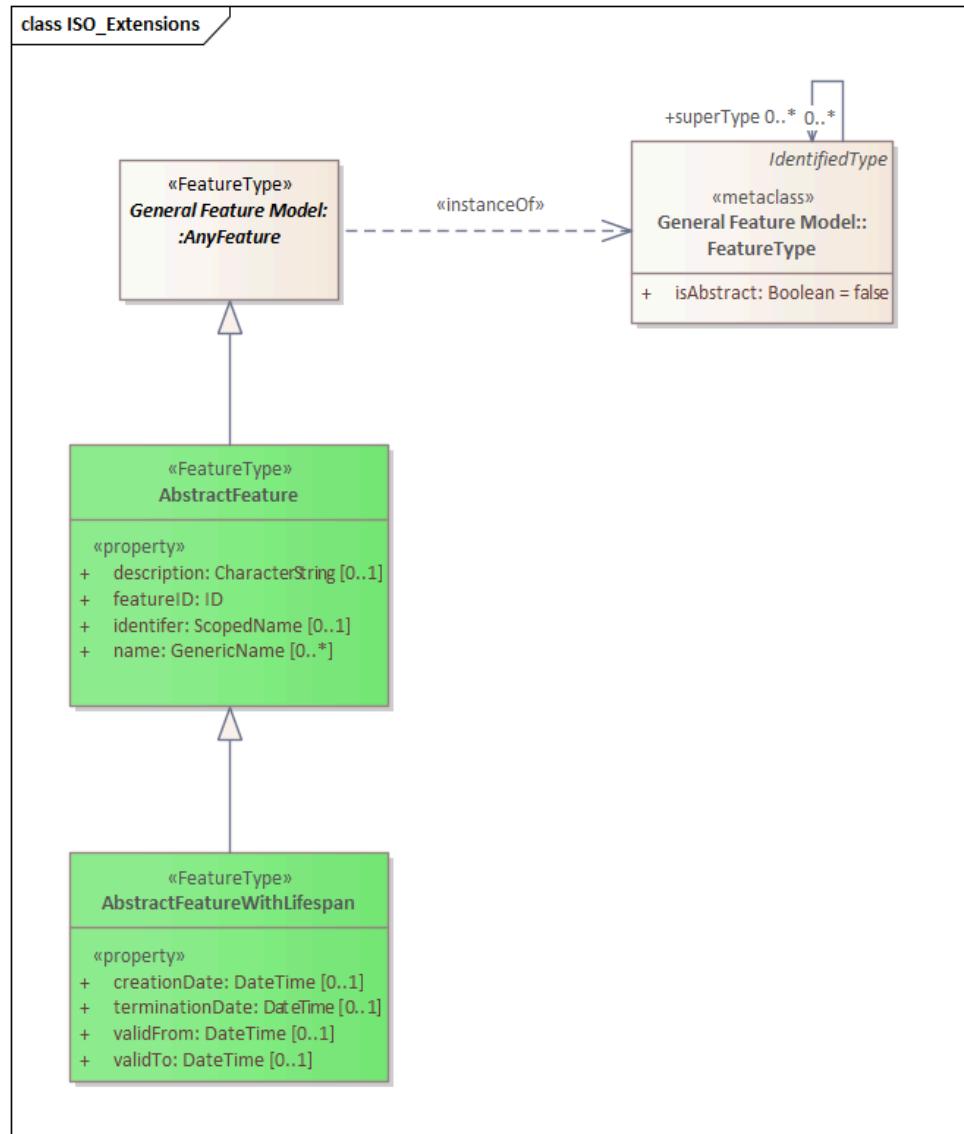


Figure 6 – POI UML Model - ISO Extensions

These extensions include further refinement of the AnyFeature class through the addition of identification and temporal validity attributes.

AbstractFeature: The root Feature class for this standard. This class has been borrowed from the CityGML 3.0 Conceptual Model.

AbstractFeatureWithLifespan: Adds temporality to AbstractFeature. This class was borrowed from the CityGML 3.0 Conceptual Model.

6.4. POI Class Model

The following classes form the core of the POI model. These classes should be the same for all POIs.

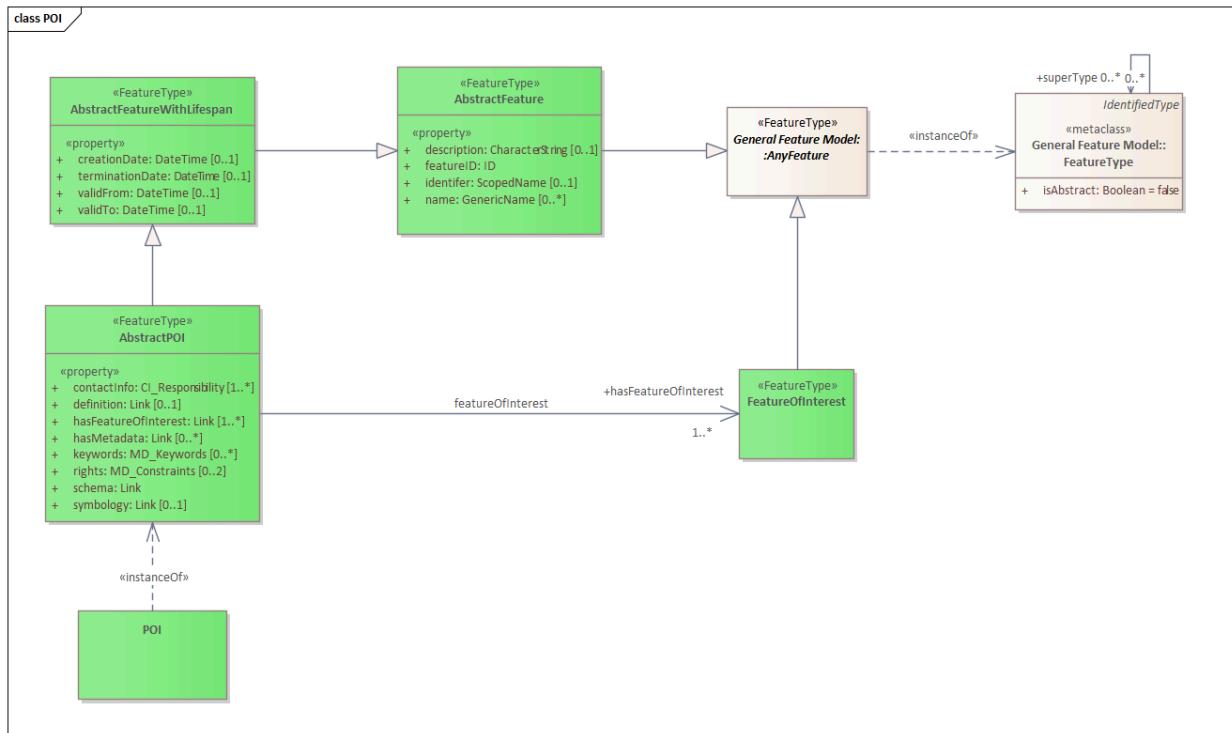


Figure 7 – POI UML Model - Core

AbstractPOI: The abstract model for a Point of Interest. All POI instances will contain these attributes.

POI: A POI instance.

FeatureOfInterest: This is an OGC Feature which has been defined independently from the POI. Conceptually, the purpose of the POI is to provide a user friendly synopsis of this Feature.

6.5. POI Payload

A POI is a representation of a Feature. The POI class provides a standard way to identify and manage a POI. However, it does not provide any information about the Feature it is representing. That is difficult to standardize since it is dependent on the data model of the Feature store being described.

Therefore, the POI model is designed to be extended with properties specific to a Feature or a Feature Collection. The POI Payload is a container for representations of Feature properties. The

syntax of those representations is provided by the Payload Schema class. Where appropriate, the semantics can also be provided through the Payload Definition class. Since the schema and definitions may be the same for a large number of Features, these classes should be instantiated as referenceable resources, allowing one instance to be used by a number of POIs.

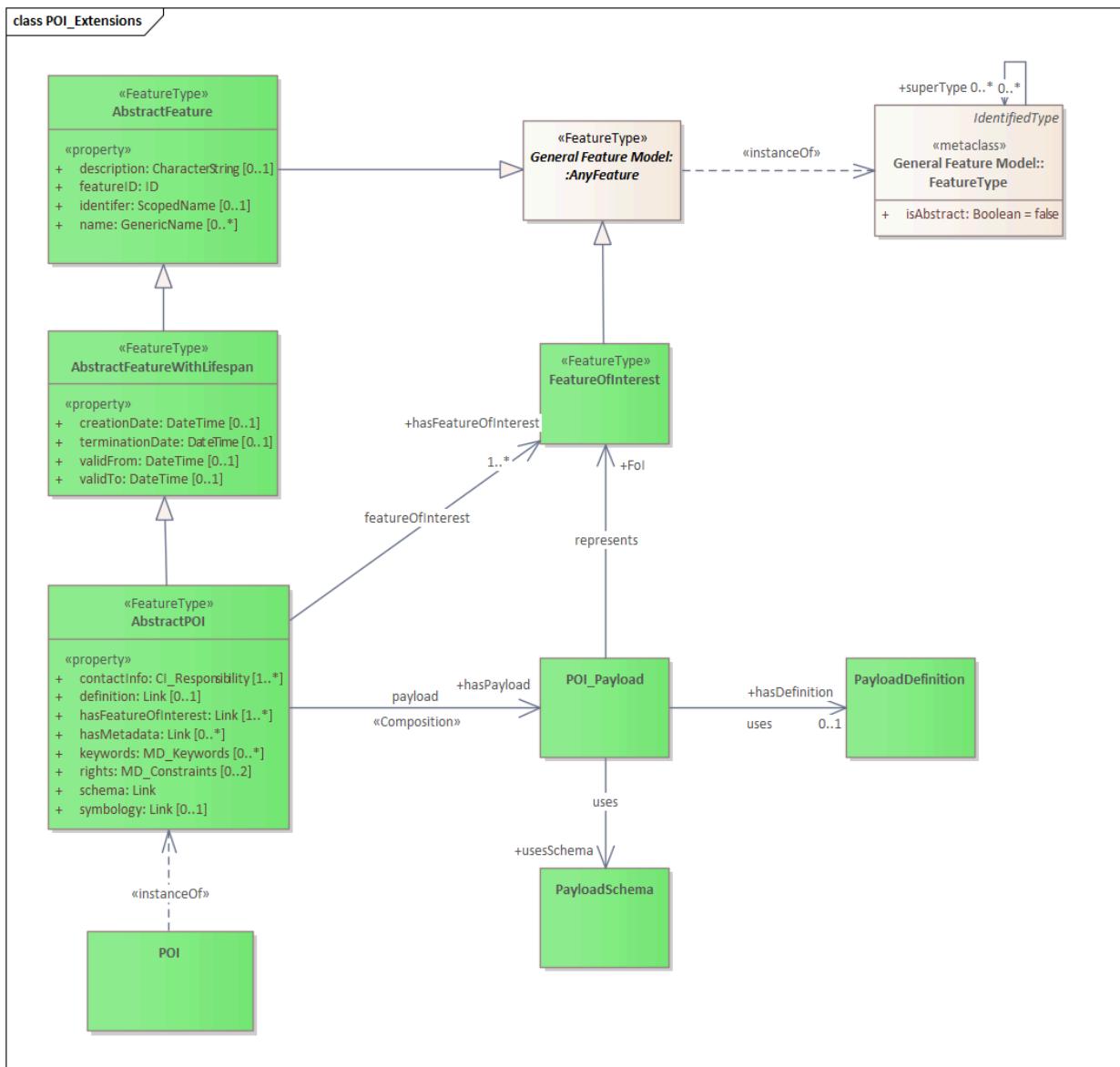


Figure 8 – POI UML Model - Payload

POIPayload: The abstract model for a Point of Interest. All POI instances will contain these attributes.

PayloadSchema: The Payload Schema Class represents a syntactic model (schema) for a POI payload.

PayloadDefinition: The Payload Definition Class represents a semantic model (ontology) for a POI payload.

6.6. POI Data Dictionary

The POI UML model is the normative definition of the POI Conceptual Model. The Data Dictionary tables in this section were software generated from the UML model. As such, this section provides a normative representation of the POI Conceptual Model.

Table 1 – Abstract Feature Class

AbstractFeature

Definition:	AbstractFeature is the abstract superclass of all feature types within the Pol Model.
Subclass of:	AnyFeature
Stereotype:	«FeatureType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
description «property»	CharacterString [0..1]	Provides further information on the feature.
featureID «property»	ID [1..1]	Specifies the unique identifier of the feature that is valid in the instance document within which it occurs.
identifier «property»	ScopedName [0..1]	Specifies the unique identifier of the feature that is valid globally.
name «property»	GenericName [0..*]	Specifies the name of the feature.

REQUIREMENT 3: REQUIREMENT – ABSTRACT FEATURE

/req/core/req-abstractfeature

- | | |
|---|---|
| A | An encoding of the AbstractFeature class SHALL be a compliant extension of the AnyFeature class defined in ISO 19109. |
| B | An encoding of the AbstractFeature class SHALL comply with requirement /req/core/req-abstractfeature-description. |
| C | An encoding of the AbstractFeature class SHALL comply with requirement /req/core/req-abstractfeature-featureid. |
| D | An encoding of the AbstractFeature class SHALL comply with requirement /req/core/req-abstractfeature-identifier. |

REQUIREMENT 3: REQUIREMENT – ABSTRACT FEATURE

E

An encoding of the AbstractFeature class SHALL comply with requirement /req/core/req-abstractfeature-name.

REQUIREMENT 4: REQUIREMENT – ABSTRACT FEATURE DESCRIPTION

/req/core/req-abstractfeature-description

A

An encoding of the AbstractFeature class SHALL include zero or one description attributes.

B

Encodings of the description attribute SHALL be a valid implementation of the CharacterString class from ISO 19103.

REQUIREMENT 5: REQUIREMENT – ABSTRACT FEATURE FEATURE ID

/req/core/req-abstractfeature-featureid

A

An encoding of the AbstractFeature class SHALL include one featureID attributes.

B

Encodings of the featureID attribute SHALL be a valid implementation of the ID class from ISO 19103.

REQUIREMENT 6: REQUIREMENT – ABSTRACT FEATURE IDENTIFIER

/req/core/req-abstractfeature-identifier

A

An encoding of the AbstractFeature class SHALL include zero or one identifier attributes.

B

Encodings of the identifier attribute SHALL be a valid implementation of the ScopedName class from ISO 19103.

REQUIREMENT 7: REQUIREMENT – ABSTRACT FEATURE NAME

/req/core/req-abstractfeature-name

A

An encoding of the AbstractFeature class SHALL include zero or more name attributes.

B

Encodings of the name attribute SHALL be a valid implementation of the GenericName class from ISO 19103.

Table 2 – Abstract Feature With Lifespan Class

AbstractFeatureWithLifespan

Definition:	AbstractFeatureWithLifespan is the base class for all POI features. This class allows the optional specification of the real-world and database times for the existence of each feature.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
creationDate «property»	DateTime [0..1]	Indicates the date at which a POI feature was added to the containing model.
terminationDate «property»	DateTime [0..1]	Indicates the date at which a POI feature was removed from the containing model.
validFrom «property»	DateTime [0..1]	Indicates the date at which a POI feature started to exist in the real world.
validTo «property»	DateTime [0..1]	Indicates the date at which a POI feature ceased to exist in the real world.

REQUIREMENT 8: REQUIREMENT – FEATURE WITH LIFESPAN

/req/core/req-featurewithlifespan

-
- A An encoding of the AbstractFeatureWithLifespan class SHALL comply with requirement /req/core/req-abstractfeature.
 - B An encoding of the AbstractFeatureWithLifespan class SHALL comply with requirement /req/core/req-featurewithlifespan-creationdate.
 - C An encoding of the AbstractFeatureWithLifespan class SHALL comply with requirement /req/core/req-featurewithlifespan-terminationdate.
 - D An encoding of the AbstractFeatureWithLifespan class SHALL comply with requirement /req/core/req-featurewithlifespan-validfrom.
 - E An encoding of the AbstractFeatureWithLifespan class SHALL comply with requirement /req/core/req-featurewithlifespan-validto.
-

REQUIREMENT 9: REQUIREMENT – FEATURE WITH LIFESPAN CREATION DATE

/req/core/req-featurewithlifespan-creationdate

-
- A An encoding of the AbstractFeatureWithLifespan class SHALL include zero or one creationDate attributes.
-

REQUIREMENT 9: REQUIREMENT – FEATURE WITH LIFESPAN CREATION DATE

- B Encodings of the creationDate attribute SHALL be a valid implementation of the DateTime class from ISO 19103.

REQUIREMENT 10: REQUIREMENT – FEATURE WITH LIFESPAN TERMINATION DATE

/req/core/req-featurewithlifespan-terminationdate

- A An encoding of the AbstractFeatureWithLifespan class SHALL include zero or one terminationDate attributes.
- B Encodings of the terminationDate attribute SHALL be a valid implementation of the DateTime class from ISO 19103.

REQUIREMENT 11: REQUIREMENT – FEATURE WITH LIFESPAN VALID FROM

/req/core/req-featurewithlifespan-validfrom

- A An encoding of the AbstractFeatureWithLifespan class SHALL include zero or one validFrom attributes.
- B Encodings of the validFrom attribute SHALL be a valid implementation of the DateTime class from ISO 19103.

REQUIREMENT 12: REQUIREMENT – FEATURE WITH LIFESPAN VALID TO

/req/core/req-featurewithlifespan-validto

- A An encoding of the AbstractFeatureWithLifespan class SHALL include zero or one validTo attributes.
- B Encodings of the validTo attribute SHALL be a valid implementation of the DateTime class from ISO 19103.

Table 3 – Abstract POI Class

AbstractPOI

Definition:	A Point of Interest (POI) is a Feature which provides a concise summary of one or more associated Features. Its purpose is to provide easy access to key information about one or more real-world objects without the need to access or understand the underlying Feature data set.
Subclass of:	AbstractFeatureWithLifespan
Stereotype:	«FeatureType»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
hasPayload	POI_Payload [0..*]	An association with a representations of zero or more properties of a Feature of Interest. This association allows Feature Property values from the Feature of Interest to be included in a POI using a standard representation.
hasFeatureOfInterest	FeatureOfInterest [1..*]	One or more Features which are represented by this POI.
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
contactInfo «property»	CI_Responsibility [1..*]	Contact information for the creators and maintainers of this POI.
definition «property»	Link [0..1]	A reference to the semantic model of the payload for this POI. The target of this reference is an instance of the PayloadDefinition class.
hasFeatureOfInterest «property»	Link [1..*]	One or more Features which are represented by this POI.
hasMetadata «property»	Link [0..*]	An association with zero or more metadata records providing additional information about this POI and/or the associated Features of Interest.
keywords «property»	MD_Keywords [0..*]	Keywords used to aid in discovery of POIs of interest.
rights «property»	MD_Constraints [0..2]	Legal and security constraints applicable to this POI.
schema «property»	Link [1..1]	A reference to the schema for the payload of this POI. The target of this reference is an instance of the PayloadSchema class.
symbology «property»	Link [0..1]	A reference to information about rendering this POI.

REQUIREMENT 13: REQUIREMENT – POI CLASS

/req/core/req-poi-class

A	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-feature-with-lifespan.
B	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-contactInfo.
C	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-definition.
D	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-featureOfInterest.

REQUIREMENT 13: REQUIREMENT – POI CLASS

E	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-metadata.
F	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-keywords.
G	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-rights.
H	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-schema.
I	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-symbology.
J	An encoding of the POI class SHALL comply with requirement / req/core/req-poi-payload.

REQUIREMENT 14: REQUIREMENT – POI CONTACT INFORMATION

/req/core/req-poi-contactInfo

A	An encoding of the POI class SHALL include one or more contactInfo attributes.
B	Encodings of the contactInfo attribute SHALL be a valid implementation of the CI_Responsibility class from ISO 19115-1: 2014

REQUIREMENT 15: REQUIREMENT – POI DEFINITION

/req/core/req-poi-definition

A	An encoding of the POI class SHALL include no more than one definition attributes.
B	Encodings of the definition attribute SHALL be a valid implementation of the Link class from this Standard.

REQUIREMENT 16: REQUIREMENT – POI FEATURE OF INTEREST

/req/core/req-poi-featureOfInterest

A	An encoding of the POI class SHALL include one or more featureOfInterest attributes.
B	Encodings of the featureOfInterest attribute SHALL be a resolvable Uniform Resource Identifier (URI) for a Feature resource.

REQUIREMENT 17: REQUIREMENT – POI METADATA

/req/core/req-poi-metadata

-
- | | |
|----------|--|
| A | An encoding of the POI class SHALL include zero or more metadata attributes. |
| B | Encodings of the metadata attribute SHALL be a resolvable Uniform Resource Identifier (URI) for a Metadata resource. |
-

REQUIREMENT 18: REQUIREMENT – POI PAYLOAD

/req/core/req-poi-payload

-
- | | |
|----------|--|
| A | An encoding of the POI class SHALL be associated with payload attributes. |
| B | Encodings of the property attribute SHALL be a valid implementation of the POIProperty class from this standard. |
-

REQUIREMENT 19: REQUIREMENT – POI KEYWORDS

/req/core/req-poi-keywords

-
- | | |
|----------|---|
| A | An encoding of the POI class SHALL include zero or more keyword attributes. |
| B | Encodings of the keyword attribute SHALL be a valid implementation of the MD_Keyword class from ISO 19115-1: 2014 |
-

REQUIREMENT 20: REQUIREMENT – POI RIGHTS

/req/core/req-poi-rights

-
- | | |
|----------|--|
| A | An encoding of the POI class SHALL include zero, one, or two rights attributes. |
| B | Encodings of the rights attribute SHALL be a valid implementation of the MD_Constraints class from ISO 19115-1: 2014 |
-

REQUIREMENT 21: REQUIREMENT – POI SCHEMA

/req/core/req-poi-schema

-
- | | |
|----------|---|
| A | An encoding of the POI class SHALL include one schema attributes. |
|----------|---|
-

REQUIREMENT 21: REQUIREMENT – POI SCHEMA

- B Encodings of the schema attribute SHALL be a resolvable Uniform Resource Identifier (URI) for a PayloadSchema resource.

REQUIREMENT 22: REQUIREMENT – POI SYMBOLOLOGY

/req/core/req-poi-symbology

- A An encoding of the POI class SHALL include zero or one symbology attributes.
- B Encodings of the symbology attribute SHALL be a resolvable Uniform Resource Identifier (URI) for a symbology resource.

Table 4 – Feature of Interest Class

FeatureOfInterest

Definition:	The thing whose property is being estimated or calculated in the course of an Observation to arrive at a Result, or whose property is being manipulated by an Actuator, or which is being sampled or transformed in an act of Sampling (SOSA).
Subclass of:	AnyFeature
Stereotype:	«FeatureType»

REQUIREMENT 23: REQUIREMENT FEATURE OF INTEREST

/req/core/req-feature-of-interest

- A An encoding of the FeatureOfInterest class SHALL be a compliant extension of the AnyFeature class defined in ISO 19109.

Table 5 – POI Payload Class

POI_Payload

Definition:	A POI_Payload provides a representation of one or more Properties of the Feature of Interest which are to be included in the POI.
Subclass of:	None
Stereotype:	«DataType»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
FoI	FeatureOfInterest [1..1]	An association with the Feature of Interest which is represented by this POI Payload.
hasDefinition	PayloadDefinition [1..0]	An association with a semantic model of this POI Payload.
usesSchema	PayloadSchema [1..1]	An association with a syntactic model (schema) of this POI Payload.

REQUIREMENT 24: REQUIREMENT – POI-PAYLOAD

/req/core/req-poi_payload-class

A	An encoding of the POI_Payload class SHALL represent zero or more Properties from the Feature of Interest that it represents.
B	The syntax of a POI_Payload class SHALL be provided in an associated instance of the PayloadSchema class.
C	The semantics of a POI_Payload class SHALL be provided in no more than one associated instance of the PayloadDefinition class.

Table 6 – Payload Definition Class

PayloadDefinition

Definition:	The Payload Definition Class represents a semantic model (ontology) for a POI payload.
Subclass of:	None
Stereotype:	None

Table 7 – Payload Schema Class

PayloadSchema

Definition:	The Payload Schema Class represents a syntactic model (schema) for a POI payload.
Subclass of:	None
Stereotype:	None

Table 8 – Link Class

Link

Definition:	A conceptual model based on the xlink construct.
Subclass of:	None
Stereotype:	None

7

MEDIA TYPES FOR ANY DATA ENCODING(S)

MEDIA TYPES FOR ANY DATA ENCODING(S)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in <http://www.iana.org/assignments/media-types/index.html> then this section may be used to define a new MIME type for registration with IANA.



A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

ANNEX A (INFORMATIVE)

CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

NOTE: Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

REQUIREMENT A.1

/req/req-class-a/req-name-1

Test purpose	Verify that...
Test method	Inspect...

A.1.2. Requirement 2



B

ANNEX B (INFORMATIVE) ISO DATA DICTIONARY

ANNEX B (INFORMATIVE) ISO DATA DICTIONARY

ISO Technical Committee 211 maintains a harmonized UML model which covers many of their standards. All of the TC211 Standards which are relevant to the POI Standard are included. Therefore the full UML model for POI consists of the classes defined in the POI UML model as well as those which referenced from the TC211 Harmonized UML model.

The Data Dictionary tables in this section were software generated from the TC211 Harmonized UML model. As such, this section provides a normative representation of the TC211 classes which are leveraged by the POI Conceptual Model.

B.1. General Feature Model

The following classes are defined in (ISO 19109:2015)

Table B.1 – Any Feature Class

AnyFeature

Definition: The class AnyFeature is an instance of the «metaclass» FeatureType (ISO 19109). It represents the set of all classes which are feature types.

+ In an implementation this abstract class shall be substituted by a concrete class representing a feature type from an application schema associated with a domain of discourse (ISO 19109, ISO 19101).

Stereotype: «FeatureType»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
-----------	-------------------------------------	------------

FeatureType [1..1]

Table B.2 – Attribute Type Class**AttributeType**

Definition:	feature attribute characteristic of a feature NOTE: A feature attribute has a name, a data type, and a value domain associated to it. A feature attribute for a feature instance also has an attribute value taken from the value domain. EXAMPLE: A feature attribute named 'color' may have an attribute value 'green' which belongs to the data type 'text'. A feature attribute named 'length' may have an attribute value '82.4' which belongs to the data type 'real'.+
Subclass Of:	PropertyType
StereoType:	«Metaclass»
Constraint:	name is mandatory (Invariant):

[cols="15,20,60",frame=none,grid=none,options="header"]<o:p></o:p>

Role name	Target class and multiplicity	Definition
featureAttributeMetadata	MD_Metadata [0..*]	
featureAttributeMetadata	MD_Metadata [0..*]	
characterizeBy	AttributeType [0..*]	
featureAttributeMetadata	MD_Metadata [0..*]	
ATTRIBUTE VALUE TYPE AND MULTIPLICITY DEFINITION		
cardinality	Multiplicity [1..1]	
valueDomain	CharacterString [1..1]	
valueType	TypeName [1..1]	

Table B.3 – Feature Type Class

FeatureType

Definition:	feature: abstraction of real world phenomena NOTE: A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant. This class describes how a feature class shall be constructed in an Application Schema. In accordance with the conformance clause of the standard, instances of this class are instantiated as feature classes in an Application Schema
Subclass Of:	IdentifiedType
Stereotype:	«Metaclass»
Constraint:	name is mandatory (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
NS_AvoidList [0..*]		
superType	FeatureType [0..*]	
featureType Metadata	MD_Metadata [0..*]	
carrier OfCharacterist	.PropertyType [0..*]	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
isAbstract	Boolean	

B.2. Geometry

The following classes are defined in (ISO 19107:2003)

Table B.4 – GM_Object Class**GM_Object**

Definition:	GM_Object is the root class of the geometric object taxonomy and supports interfaces common to all geographically referenced geometric objects. GM_Object instances are sets of direct positions in a particular coordinate reference system. A GM_Object can be regarded as an infinite set of points that satisfies the set operation interfaces for a set of direct positions, TransfiniteSet<DirectPosition>. Since an infinite collection class cannot be implemented directly, a Boolean test for inclusion shall be provided by the GM_Object interface. This international standard concentrates on vector geometry classes, but future work may use GM_Object as a root class without modification. NOTE As a type, GM_Object does not have a well-defined default state or value representation as a data type. Instantiated subclasses of GM_Object will.
Subclass Of:	none
Stereotype:	«type»
Constraint:	dimension() > boundary().dimension (Invariant):
Constraint:	boundary().notEmpty() implies boundary().dimension() = dimension() -1 (Invariant):
Constraint:	boundary().isEmpty() = isCycle() (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	Geometry [1..1]	
	Transfinite Set<Direct Position> [1..1]	
CRS	CV_DomainObject [1..1]	
CRS	CRS [0..1]	
CRS	SC_CRS [0..1]	

Table B.5 – GM_Point Class**GM_Point**

Definition:	GM_Point is the basic data type for a geometric object consisting of one and only one point.
Subclass Of:	GM_Primitive
Stereotype:	«type»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
composite	Point [1..1]	
	GM_Composite Point [0..*]	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
position	DirectPosition [1..1]	<p>The attribute “position” shall be the DirectPosition of this GM_Point. GM_Point::position [1] : DirectPosition NOTE In most cases, the state of a GM_Point is fully determined by its position attribute. The only exception to this is if the GM_Point has been subclassed to provide additional non-geometric information such as symbology.</p>

Table B.6 – GM_LineString Class**GM_LineString**

Definition:	A GM_LineString (Figure 16) consists of sequence of line segments, each having a parameterization like the one for GM_LineSegment (See 6.4.11). The class essentially combines a Sequence<GM_LineSegments> into a single object, with the obvious savings of storage space.
Subclass Of:	GM_Primitive
Stereotype:	«type»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
controlPoint	GM_PointArray [1..1]	

Table B.7 – GM_Polygon Class

GM_Polygon

Definition:	A GM_Polygon (Figure 21) is a surface patch that is defined by a set of boundary curves and an underlying surface to which these curves adhere. The default is that the curves are coplanar and the polygon uses planar interpolation in its interior.
Subclass Of:	GM_Primitive
Stereotype:	«type»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
surface	GM_Polyhedral Surface [0..1]	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
boundary	GM_Surface Boundary	
spanning Surface	GM_Surface [0..1]	

B.3. Citation and responsible party information

The following classes are defined in (ISO 19115-1 Edition 1)

Table B.8 – CI_Address Class

CI_Address

Definition: location of the responsible individual or organisation
 StereoType: None

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
administrativeArea	CharacterString [0..1]	state, province of the location
city	CharacterString [0..1]	city of the location
country	CharacterString [0..1]	country of the physical address
deliveryPoint	CharacterString [0..*]	address line for the location Example Street number and name, suite number, etc
electronicMailAddress	CharacterString [0..*]	address of the electronic mailbox of the responsible organisation or individual
postalCode	CharacterString [0..1]	ZIP or other postal code

Table B.9 – CI_Citation Class

CI_Citation

Definition: standardized resource reference
 StereoType: None

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
alternateTitle	CharacterString [0..*]	short name or other language name by which the cited information is known. Example: DCW as an alternative title for Digital Chart of the World
cited Responsible Party	CI_Responsibility [0..*]	roles, name, contact, and position information for an individual or organisation that is responsible for the resource
date	CI_Date [0..*]	reference date for the cited resource
edition	CharacterString [0..1]	version of the cited resource

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
editionDate	DateTime [0..1]	date of the edition
graphic	MD_BrowseGraphic [0..*]	citation graphic or logo for cited party
identifier	MD_Identifier [0..*]	value uniquely identifying an object within a namespace
ISBN	CharacterString [0..1]	international Standard Book Number
ISSN	CharacterString [0..1]	international Standard Serial Number
online Resource	CI_OnlineResource [0..*]	online reference to the cited resource
otherCitation Details	CharacterString [0..*]	other information required to complete the citation that is not recorded elsewhere
presentation Form	CI_PresentationFormCode [0..*]	mode in which the resource is represented
series	CI_Series [0..1]	information about the series, or aggregate resource, of which the resource is a part
title	CharacterString [1..1]	name by which the cited resource is known

Table B.10 – CI_Contact Class

CI_Contact

Definition:	information required to enable contact with the responsible person and/or organisation
Stereotype:	None

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
address	CI_Address [0..*]	physical and email address at which the organisation or individual may be contacted
contact Instructions	CharacterString [0..1]	supplemental instructions on how or when to contact the individual or organisation
contactType	CharacterString [0..1]	type of contact

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
hours OfService	CharacterString [0..*]	time period (including time zone) when individuals can contact the organisation or individual
online Resource	CI_OnlineResource [0..*]	on-line information that can be used to contact the individual or organisation
phone	CI_Telephone [0..*]	telephone numbers at which the organisation or individual may be contacted

Table B.11 – CI_Date Class

CI_Date

Definition: reference date and event used to describe it

Stereotype: «DataType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
date	DateTime [1..1]	reference date for the cited resource
dateType	CI_DateTypeCode [1..1]	event used for reference date

Table B.12 – CI_DateTypeCode Codelist

CI_DateTypeCode

Definition: identification of when a given event occurred

Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
adopted	literal	date identifies when resource was adopted
creation	literal	date identifies when the resource was brought into existence

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
deprecated	literal	date identifies when resource was deprecated
distribution	literal	date identifies when an instance of the resource was distributed
expiry	literal	date identifies when resource expires
inForce	literal	date identifies when resource became in force
lastRevision	literal	date identifies when resource was last reviewed
lastUpdate	literal	date identifies when resource was last updated
nextUpdate	literal	date identifies when resource will be next updated
publication	literal	date identifies when the resource was issued
released	literal	the date that the resource shall be released for public access
revision	literal	date identifies when the resource was examined or re-examined and improved or amended
superseded	literal	date identifies when resource was superseded or replaced by another resource
unavailable	literal	date identifies when resource became not available or obtainable
validityBegins	literal	time at which the data is considered to become valid. Note: There could be quite a delay between creation and validity begins
validityExpires	literal	time at which the data is no longer considered to be valid

Table B.13 – CI_Individual Class

CI_Individual

Definition: information about the party if the party is an individual

Subclass Of: CI_Party

Stereotype: None

Constraint: count (name + positionName) > 0 (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	CI_Organisation [1..1]	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
positionName	CharacterString [0..1]	position of the individual in an organisation

Table B.14 – CI_OnLineFunctionCode Codelist

CI_OnLineFunctionCode

Definition: function performed by the resource

Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
browseGraphic	literal	browse graphic provided
browsing	literal	online browsing provided
completeMetadata	literal	complete metadata provided
download	literal	online instructions for transferring data from one storage device or system to another
emailService	literal	online email service provided
fileAccess	literal	online file access provided
information	literal	online information about the resource
offlineAccess	literal	online instructions for requesting the resource from the provider

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
order	literal	online order process for obtaining the resource
search	literal	online search interface for seeking out information about the resource
upload	literal	online resource upload capability provided

Table B.15 – CI_OnlineResource Class

CI_OnlineResource

Definition:	information about on-line sources from which the resource, specification, or community profile name and extended metadata elements can be obtained
Stereotype:	«DataType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
application Profile	CharacterString [0..1]	name of an application profile that can be used with the online resource
description	CharacterString [0..1]	detailed text description of what the online resource is/ does
function	CI_OnLineFunctionCode [0..1]	code for function performed by the online resource
linkage	CharacterString [1..1]	location (address) for on-line access using a Uniform Resource Locator/Uniform Resource Identifier address or similar addressing scheme such as http://www.statkart.no/isotc211
name	CharacterString [0..1]	name of the online resource
protocol	CharacterString [0..1]	connection protocol to be used e.g. http, ftp, file, http get KVP, http POST, etc...
protocol Request	CharacterString [0..1]	protocol used by the accessed resource(to be used mainly for POST requests). Example POST/XML: <GetFeature service="WFS" version="2.0.0" outputFormat="application/gml+xml; version=3.2" xmlns=http://www.opengis.net/wfs/2.0 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation="http://www.opengis.net/

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
		wfs/2.0http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"> <Query typeNames="Roads"/> </GetFeature>

Table B.16 – CI_Organisation Class

CI_Organisation

Definition:	information about the party if the party is an organisation
Subclass Of:	CI_Party
Stereotype:	None
Constraint:	count (name + logo) > 0 (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
individual	CI_Individual [0..*]	an individual in the named organisation

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
logo	MD_Browse Graphic [0..*]	Graphic identifying organization

Table B.17 – CI_Party Class

CI_Party

Definition:	information about the individual and/or organisation of the party
Stereotype:	«abstract»

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	CI_Responsibility []	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
contactInfo	CI_Contact [0..*]	contact information for the party
name	CharacterString [0..1]	name of the party (individual or organization)

Table B.18 – CI_PresentationFormCode Codelist

CI_PresentationFormCode

Definition:	mode in which the data is represented
Stereotype:	«CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
audioDigital	literal	digital audio recording
audio Hardcopy	literal	audio recording delivered by analog media, such as a magnetic tape
diagram Digital	literal	information represented graphically by charts such as pie chart, bar chart, and other type of diagrams and recorded in digital format
diagram Hardcopy	literal	information represented graphically by charts such as pie chart, bar chart, and other type of diagrams and printed on paper, photographic material, or other media
document Digital	literal	digital representation of a primarily textual item (can contain illustrations also)
document Hardcopy	literal	representation of a primarily textual item (can contain illustrations also) on paper, photographic material, or other media
imageDigital	literal	likeness of natural or man-made features, objects, and activities acquired through the sensing of visual or any other segment of the electromagnetic spectrum by sensors,

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
image Hardcopy	literal	such as thermal infrared, and high resolution radar and stored in digital format
mapDigital	literal	likeness of natural or man-made features, objects, and activities acquired through the sensing of visual or any other segment of the electromagnetic spectrum by sensors, such as thermal infrared, and high resolution radar and reproduced on paper, photographic material, or other media for use directly by the human user
mapHardcopy	literal	map represented in raster or vector form
modelDigital	literal	map printed on paper, photographic material, or other media for use directly by the human user
model Hardcopy	literal	multi-dimensional digital representation of a feature, process, etc.
multimedia Digital	literal	3-dimensional, physical model
multimedia Hardcopy	literal	information representation using simultaneously various digital modes for text, sound, image
physical Object	literal	information representation using simultaneously various analog modes for text, sound, image
profileDigital	literal	a physical object. Eg. Rock or mineral sample, microscope slide
profile Hardcopy	literal	vertical cross-section in digital form
tableDigital	literal	vertical cross-section printed on paper, etc.
table Hardcopy	literal	digital representation of facts or figures systematically displayed, especially in columns
videoDigital	literal	representation of facts or figures systematically displayed, especially in columns, printed on paper, photographic material, or other media
video Hardcopy	literal	digital video recording
		video recording on film

Table B.19 – CI_Responsibility Class

CI_Responsibility

Definition: information about the party and their role

Stereotype: None

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
party	CI_Party [1..*]	information about the party

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
extent	EX_Extent [0..*]	spatial or temporal extent of the role
role	CI_RoleCode [1..1]	function performed by the responsible party

Table B.20 – CI_RoleCode Codelist

CI_RoleCode

Definition: function performed by the responsible party

Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
author	literal	party who authored the resource
coAuthor	literal	party who jointly authors the resource
collaborator	literal	party who assists with the generation of the resource other than the principal investigator
contributor	literal	party contributing to the resource
custodian	literal	party that accepts accountability and responsibility for the resource and ensures appropriate care and maintenance of the resource
distributor	literal	party who distributes the resource

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
editor	literal	party who reviewed or modified the resource to improve the content
funder	literal	party providing monetary support for the resource
mediator	literal	a class of entity that mediates access to the resource and for whom the resource is intended or useful
originator	literal	party who created the resource
owner	literal	party that owns the resource
point OfContact	literal	party who can be contacted for acquiring knowledge about or acquisition of the resource
principal Investigator	literal	key party responsible for gathering information and conducting research
processor	literal	party who has processed the data in a manner such that the resource has been modified
publisher	literal	party who published the resource
resource Provider	literal	party that supplies the resource
rightsHolder	literal	party owning or managing rights over the resource
sponsor	literal	party who speaks for the resource
stakeholder	literal	party who has an interest in the resource or the use of the resource
user	literal	party who uses the resource

Table B.21 – CI_Series Class

CI_Series

Definition:	information about the series, or aggregate resource, to which a resource belongs
Stereotype:	«DataType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
issue Identification	CharacterString [0..1]	information identifying the issue of the series
name	CharacterString [0..1]	name of the series, or aggregate resource, of which the resource is a part
page	CharacterString [0..1]	details on which pages of the publication the article was published

Table B.22 – CI_Telephone Class

CI_Telephone

Definition: telephone numbers for contacting the responsible individual or organisation

Stereotype: «DataType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
number	CharacterString [1..1]	telephone number by which individuals can contact responsible organisation or individual
numberType	CI_TelephoneTypeCode [0..1]	type of telephone responsible organisation or individual

Table B.23 – CI_TelephoneTypeCode Codelist

CI_TelephoneTypeCode

Definition: type of telephone

Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
facsimile	literal	telephone provides facsimile service
sms	literal	telephone provides sms service

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
voice	literal	telephone provides voice service

B.4. Constraint information

The following classes are defined in (ISO 19115-1 Edition 1)

Table B.24 – MD_Classification Class

MD_ClassificationCode

Definition:	name of the handling restrictions on the resource
Stereotype:	«CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
confidential	literal	available for someone who can be entrusted with information
forOfficial UseOnly	literal	unclassified information that may be exempt from mandatory release to the public
limited Distribution	literal	desimination limited by designating body
protected	literal	compromise of the information could cause damage
restricted	literal	not for general disclosure
secret	literal	kept or meant to be kept private, unknown, or hidden from all but a select group of people
sensitive ButUnclassified	literal	although unclassified, requires strict controls over its distribution
topSecret	literal	of the highest secrecy
unclassified	literal	available for general disclosure

Table B.25 – MD_Constraints Class**MD_Constraints**

TARGET CLASS AND MULTIPLICITY		
ROLE NAME	DEFINITION	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
	MD_Identification []	
	MD_Metadata []	
constraint Application Scope	MD_Scope [0..1]	Spatial and temporal extent of the application of the constraint restrictions
graphic	MD_Browse Graphic [0..*]	graphic /symbol indicating the constraint
reference	CI_Citation [0..*]	citation/URL for the limitation or constraint, eg. copyright statement, license agreement, etc
releasability	MD_Releasability [0..1]	information concerning the parties to whom the resource can or cannot be released
responsible Party	CI_Responsibility [0..*]	party responsible for the resource constraints
useLimitation	CharacterString [0..*]	limitation affecting the fitness for use of the resource or metadata. Example, “not to be used for navigation”

Table B.26 – MD_LegalConstraints Class**MD_LegalConstraints**

Definition:	restrictions and legal prerequisites for accessing and using the resource or metadata
Subclass Of:	MD_Constraints

Stereotype:	None
Constraint:	otherConstraints: only documented if accessConstraints or useConstraints = "otherRestrictions" (Invariant): If MD_LegalConstraints used then count of (accessConstraints + use Constraints + otherConstraints + useLimitation + releasability) > 0 (Invariant):
Constraint:	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
access Constraints	MD_Restriction Code [0..*]	access constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations on obtaining the resource or metadata
other Constraints	CharacterString [0..*]	other restrictions and legal prerequisites for accessing and using the resource or metadata
use Constraints	MD_Restriction Code [0..*]	constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations or warnings on using the resource or metadata

Table B.27 – MD_Releasability Class

MD_Releasability

Definition:	information about resource release constraints
Stereotype:	None
Constraint:	count (addressee + statement) > 0 (Invariant):

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
addressee	CI_Responsibility [0..*]	party to which the release statement applies
dissemination Constraints	MD_Restriction Code [0..*]	component in determining releasability
statement	CharacterString [0..1]	release statement

Table B.28 – MD_RestrictionCode Codelist**MD_RestrictionCode**

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
confidential	literal	not available to the public contains information that could be prejudicial to a commercial, industrial, or national interest
copyright	literal	exclusive right to the publication, production, or sale of the rights to a literary, dramatic, musical, or artistic work, or to the use of a commercial print or label, granted by law for a specified period of time to an author, composer, artist, distributor
in-confidence	literal	with trust
intellectual Property Rights	literal	rights to financial benefit from and control of distribution of non-tangible property that is a result of creativity
licence	literal	formal permission to do something
licence Distributor	literal	formal permission required for a person or an entity to commercialize or distribute the resource
licence EndUser	literal	formal permission required for a person or an entity to use the resource and that may differ from the person that orders or purchases it
licence Unrestricted	literal	formal permission not required to use the resource
other Restrictions	literal	limitation not listed
patent	literal	government has granted exclusive right to make, sell, use or license an invention or discovery
patent Pending	literal	produced or sold information awaiting a patent
private	literal	protects rights of individual or organisations from observation, intrusion, or attention of others
restricted	literal	withheld from general circulation or disclosure

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
sensitive ButUnclassifie	literal	although unclassified, requires strict controls over its distribution.
statutory	literal	prescribed by law
trademark	literal	a name, symbol, or other device identifying a product, officially registered and legally restricted to the use of the owner or manufacturer
unrestricted	literal	no constraints exist

Table B.29 – MD_SecurityConstraints Class

MD_SecurityConstraints

Definition:	handling restrictions imposed on the resource or metadata for national security or similar security concerns
Subclass Of:	MD_Constraints
Stereotype:	None

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
classification	MD_Classification Code [1..1]	name of the handling restrictions on the resource or metadata
classification System	CharacterString [0..1]	name of the classification system
handling Description	CharacterString [0..1]	additional information about the restrictions on handling the resource or metadata
userNote	CharacterString [0..1]	explanation of the application of the legal constraints or other restrictions and legal prerequisites for obtaining and using the resource or metadata

B.5. Identification information

The following classes are defined in (ISO 19115-1 Edition 1)

Table B.30 – DS_AssociationTypeCode Codelist**DS_AssociationTypeCode**

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
collectiveTitle	literal	common title with holdings note NOTE: title identifies elements of a series collectively, combined with information about what volumes are available at the source cited
cross Reference	literal	reference from one resource to another
dependency	literal	associate through a dependency
isComposed Of	literal	reference to resources that are parts of this data set
largerWork Citation	literal	reference to a master resource of which this one is a part
part OfSeamless Database	literal	part of same structured set of data held in a computer
revisionOf	literal	resource is a revision of associated resource
series	literal	associated through a common heritage such as produced to a common product specification
stereoMate	literal	part of a set of imagery that when used together, provides three-dimensional images

Table B.31 – DS_InitiativeTypeCode Codelist**DS_InitiativeTypeCode**

Definition:	type of aggregation activity in which resources are related
Stereotype:	«CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
campaign	literal	series of organized planned actions
collection	literal	accumulation of resources assembled for a specific purpose
exercise	literal	specific performance of a function or group of functions
experiment	literal	process designed to find if something is effective or valid
investigation	literal	search or systematic inquiry
mission	literal	specific operation of a data collection system
operation	literal	action that is part of a series of actions
platform	literal	vehicle or other support base that holds a sensor
process	literal	method of doing something involving a number of steps
program	literal	specific planned activity
project	literal	organized undertaking, research, or development
sensor	literal	device or piece of equipment which detects or records
study	literal	examination or investigation
task	literal	piece of work
trial	literal	process of testing to discover or demonstrate something

Table B.32 – MD_AssociatedResource Class

MD_AssociatedResource

Definition: associated resource information

Subclass Of: GP_AssociatedResource

Stereotype: None

Constraint: count of (name + metadataReference) > 0 (Invariant):

TARGET CLASS AND MULTIPLICITY		
MD_Identification []		
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
associationType	DS_Association TypeCode [1..1]	type of relation between the resources
initiativeType	DS_InitiativeType Code [0..1]	type of initiative under which the associated resource was produced
metadataReference	CI_Citation [0..1]	reference to the metadata of the associated resource
name	CI_Citation [0..1]	citation information about the associated resource

Table B.33 – MD_DataIdentification Class

MD_DataIdentification

Definition:	information required to identify a resource
Subclass Of:	MD_Identification
Stereotype:	None
Constraint:	defaultLocale documented if resource includes textual information (Invariant):
Constraint:	defaultLocale.PT_Locale.characterEncoding default value is UTF-8 (Invariant):

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
defaultLocale	PT_Locale [0..1]	language and character set used within the resource
environmentDescription	CharacterString [0..1]	description of the resource in the producer's processing environment, including items such as the software, the computer operating system, file name, and the dataset size

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
otherLocale	PT_Locale [0..*]	alternate localised language(s) and character set (s) used within the resource
supplemental Information	CharacterString [0..1]	any other descriptive information about the resource

Table B.34 – MD_Identification Class

MD_Identification

Definition:	basic information required to uniquely identify a resource or resources
Subclass Of:	GP_Identification
Stereotype:	«abstract»
Constraint:	(MD_Metadata.metadataScope.MD_MetadataScope.resourceScope) = 'dataset' implies count (extent.geographicElement.EX_GeographicBoundingBox + extent.geographicElement.EX_GeographicDescription) >= 1 (Invariant);
Constraint:	(MD_Metadata.metadataScope.MD_Scope.resourceScope) = ('dataset' or 'series') implies topicCategory is mandatory (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
resource Maintenance	MD_MaintenanceInformation [0..*]	information about the frequency of resource updates, and the scope of those updates
associated Resource	MD_AssociatedResource [0..*]	associated resource information
resource SpecificUsage	MD_Usage [0..*]	basic information about specific application(s) for which the resource(s) has/have been or is being used by different users
MD_Metadata []		

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
abstract	CharacterString [1..1]	brief narrative summary of the content of the resource(s)

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
additional Documentation	CI_Citation [0..*]	other documentation associated with the resource EXAMPLE Related articles, publications, user guides, data dictionaries.
citation	CI_Citation [1..1]	citation for the resource(s)
credit	CharacterString [0..*]	recognition of those who contributed to the resource(s)
extent	EX_Extent [0..*]	spatial and temporal extent of the resource
point OfContact	CI_Responsibility [0..*]	identification of, and means of communication with, person(s) and organisation(s) associated with the resource(s)
processing Level	MD_Identifier [0..1]	code that identifies the level of processing in the producers coding system of a resource eg. NOAA level 1B
purpose	CharacterString [0..1]	summary of the intentions with which the resource(s) was developed
spatial Representation Type	MD_Spatial RepresentationTypeCode [0..*]	method used to spatially represent geographic information
spatial Resolution	MD_Resolution [0..*]	factor which provides a general understanding of the density of spatial data in the resource or describes the range of resolutions in which a digital resource may be used NOTE: this element should be repeated when describing upper and lower range
status	MD_ProgressCode [0..*]	status of the resource(s)
temporal Resolution	TM_Duration [0..*]	smallest resolvable temporal period in a resource
topicCategory	MD_TopicCategoryCode [0..*]	main theme(s) of the resource

Table B.35 – MD_KeywordClass Class

MD_KeywordClass

Definition:	specification of a class to categorize keywords in a domain-specific vocabulary that has a binding to a formal ontology
StereoType:	None

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	MD_Keywords []	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
className	CharacterString [1..1]	1. character string to label the keyword category in natural language
concept Identifier	URI [0..1]	URI of concept in ontology specified by the ontology attribute; this concept is labeled by the className: CharacterString.
ontology	CI_Citation [1..1]	a reference that binds the keyword class to a formal conceptualization of a knowledge domain for use in semantic processing NOTE: Keywords in the associated MD_Keywords keyword list must be within the scope of this ontology

Table B.36 – MD_Keywords Class

MD_Keywords

Definition:	keywords, their type and reference source NOTE: When the resource described is a service, one instance of MD_Keyword shall refer to the service taxonomy defined in ISO 19119, 8.3)
Stereotype:	None
Constraint:	When the resource described is a service, one instance of MD_Keyword shall refer to the service taxonomy defined in ISO 19119 (Invariant):

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	MD_Identification []	
keywordClass	MD_KeywordClass [0..1]	association of a MD_Keywords instance with a MD_KeywordClass to provide user-defined categorization of groups of keywords that extend or are orthogonal to the standardized KeywordTypeCodes and are associated with an ontology that allows additional semantic query processing

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
keyword	CharacterString [1..*]	1. commonly used word(s) or formalised word(s) or phrase(s) used to describe the subject
thesaurus Name	CI_Citation [0..1]	name of the formally registered thesaurus or a similar authoritative source of keywords
type	MD_KeywordTypeCode [0..1]	subject matter used to group similar keywords

Table B.37 – MD_KeywordTypeCode Codelist

MD_KeywordTypeCode

Definition:	methods used to group similar keywords
Subclass Of:	TaxonomyKeywords
Stereotype:	«CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
dataCentre	literal	keyword identifies a repository or archive that manages and distributes data
discipline	literal	keyword identifies a branch of instruction or specialized learning
featureType	literal	keyword identifies a resource containing or about a collection of feature instances with common characteristics
instrument	literal	keyword identifies a device used to measure or compare physical properties
place	literal	keyword identifies a location
platform	literal	keyword identifies a structure upon which an instrument is mounted
process	literal	keyword identifies a series of actions or natural occurrences
product	literal	keyword identifies a type of product
project	literal	keyword identifies an endeavour undertaken to create or modify a product or service

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
service	literal	keyword identifies an activity carried out by one party for the benefit of another
stratum	literal	keyword identifies the layer(s) of any deposited substance or levels within an ordered system
subTopic Category	literal	refinement of a topic category for the purpose of geographic data classification
taxon	literal	keyword identifies a taxonomy of the resource
temporal	literal	keyword identifies a time period related to the resource
theme	literal	keyword identifies a particular subject or topic

Table B.38 – MD_ProgressCode Codelist

MD_ProgressCode

Definition:	status of the resource
Stereotype:	«CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
accepted	literal	agreed to by sponsor
completed	literal	has been completed
deprecated	literal	resource superseded and will become obsolete, use only for historical purposes
final	literal	progress concluded and no changes will be accepted
historical Archive	literal	stored in an offline storage facility
notAccepted	literal	rejected by sponsor
obsolete	literal	no longer relevant
onGoing	literal	continually being updated

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
pending	literal	committed to, but not yet addressed
planned	literal	fixed date has been established upon or by which the resource will be created or updated
proposed	literal	suggested that development needs to be undertaken
required	literal	needs to be generated or updated
retired	literal	item is no longer recommended for use. It has not been superseded by another item
superseded	literal	replaced by new
tentative	literal	provisional changes likely before resource becomes final or complete
under Development	literal	currently in the process of being created
valid	literal	acceptable under specific conditions
withdrawn	literal	removed from consideration

Table B.39 – MD_REpresentativeFraction Class

MD_RepresentativeFraction

Definition:	derived from ISO 19103 Scale where MD_RepresentativeFraction. denominator = 1 / Scale.measure And Scale.targetUnits = Scale.sourceUnits
StereoType:	«DataType»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
denominator	Integer [1..1]	the number below the line in a vulgar fraction

Table B.40 – MD_Resolution Class

MD_Resolution

Definition: level of detail expressed as a scale factor, a distance or an angle

Stereotype: «Union»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
angular Distance	Angle	Angular sampling measure
distance	Distance	horizontal ground sample distance
equivalent Scale	MD_ Representative Fraction	level of detail expressed as the scale of a comparable hardcopy map or chart
levelOfDetail	CharacterString	brief textual description of the spatial resolution of the resource
vertical	Distance	Vertical sampling distance

Table B.41 – MD_SpatialRepresentationTypeCode Codelist

MD_SpatialRepresentationTypeCode

Definition: method used to represent geographic information in the resource

Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
grid	literal	grid data is used to represent geographic data
stereoModel	literal	three-dimensional view formed by the intersecting homologous rays of an overlapping pair of images
textTable	literal	textual or tabular data is used to represent geographic data
tin	literal	triangulated irregular network
vector	literal	vector data is used to represent geographic data
video	literal	scene from a video recording

Table B.42 – MD_Usage Class**MD_Usage**

Definition:	brief description of ways in which the resource(s) is/are currently or has been used
Subclass Of:	GP_Usage
Stereotype:	None

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
additional Documentation	CI_Citation [0..*]	publications that describe usage of data
identified Issues	CI_Citation [0..1]	citation of a description of known issues associated with the resource along with proposed solutions if available
response	CharacterString [0..*]	response to the user-determined limitations E.G.. ‘this has been fixed in version x’
specificUsage	CharacterString [1..1]	brief description of the resource and/or resource series usage
usageDate Time	TM_Primitive [0..*]	date and time of the first use or range of uses of the resource and/or resource series
userContact Info	CI_Responsibility [0..*]	identification of and means of communicating with person(s) and organisation(s) using the resource(s)
user Determined Limitations	CharacterString [0..1]	applications, determined by the user for which the resource and/or resource series is not suitable

Table B.43 – MD_TopicCategoryCode Codelist**MD_TopicCategoryCode**

Definition:	high-level geographic data thematic classification to assist in the grouping and search of available geographic data sets. NOTE 1 Can be used to group keywords as well. Listed examples are not exhaustive. NOTE 2: It is understood there are overlaps between general categories and the user is encouraged to select the one most appropriate.
Stereotype:	enumeration

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
farming	literal	rearing of animals and/or cultivation of plants Examples: agriculture, irrigation, aquaculture, plantations, herding, pests and diseases affecting crops and livestock
biota	literal	flora and/or fauna in natural environment Examples: wildlife, vegetation, biological sciences, ecology, wilderness, sealife, wetlands, habitat
boundaries	literal	legal land descriptions Examples: political and administrative boundaries
climatology Meteorology Atmosphere	literal	processes and phenomena of the atmosphere Examples: cloud cover, weather, climate, atmospheric conditions, climate change, precipitation
economy	literal	economic activities, conditions and employment Examples: production, labour, revenue, commerce, industry, tourism and ecotourism, forestry, fisheries, commercial or subsistence hunting, exploration and exploitation of resources such as minerals, oil and gas
elevation	literal	height above or below a vertical datum Examples: altitude, bathymetry, digital elevation models, slope, derived products
environment	literal	environmental resources, protection and conservation Examples: environmental pollution, waste storage and treatment, environmental impact assessment, monitoring environmental risk, nature reserves, landscape
geoscientific Information	literal	information pertaining to earth sciences Examples: geophysical features and processes, geology, minerals, sciences dealing with the composition, structure and origin of the earth's rocks, risks of earthquakes, volcanic activity, landslides, gravity information, soils, permafrost, hydrogeology, erosion
health	literal	health, health services, human ecology, and safety Examples: disease and illness, factors affecting health, hygiene, substance abuse, mental and physical health, health services
imageryBase MapsEarth Cover	literal	base maps Examples: land cover, topographic maps, imagery, unclassified images, annotations
intelligence Military	literal	military bases, structures, activities Examples: barracks, training grounds, military transportation, information collection
inlandWaters	literal	inland water features, drainage systems and their characteristics Examples: rivers and glaciers, salt lakes,

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
		water utilization plans, dams, currents, floods, water quality, hydrographic charts
location	literal	positional information and services Examples: addresses, geodetic networks, control points, postal zones and services, place names
oceans	literal	features and characteristics of salt water bodies (excluding inland waters) Examples: tides, tidal waves, coastal information, reefs
planning Cadastre	literal	information used for appropriate actions for future use of the land Examples: land use maps, zoning maps, cadastral surveys, land ownership
society	literal	characteristics of society and cultures Examples: settlements, anthropology, archaeology, education, traditional beliefs, manners and customs, demographic data, recreational areas and activities, social impact assessments, crime and justice, census information
structure	literal	man-made construction Examples: buildings, museums, churches, factories, housing, monuments, shops, towers
transportation	literal	means and aids for conveying persons and/or goods Examples: roads, airports/airstrips, shipping routes, tunnels, nautical charts, vehicle or vessel location, aeronautical charts, railways
utilities Communication	literal	energy, water and waste systems and communications infrastructure and services Examples: hydroelectricity, geothermal, solar and nuclear sources of energy, water purification and distribution, sewage collection and disposal, electricity and gas distribution, data communication, telecommunication, radio, communication networks
extra Terrestrial	literal	region more than 100 km above the surface of the Earth
disaster	literal	Information related to disasters Examples: site of the disaster, evacuation zone, disaster-prevention facility, disaster relief activities

B.6. Name types

The following classes are defined in (ISO 19103:2015)

Table B.44 – Generic Name Class

GenericName

Definition:	Generic Name is the abstract class for all names in a NameSpace. Each instance of a GenericName is either a LocalName or a ScopedName. A LocalName references a local object directly accessible from the Name Space. A ScopedName is a composite of a LocalName for locating another NameSpace and a GenericName valid in that NameSpace.
Stereotype:	interface

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
scope	NameSpace [1..1]	

Table B.45 – Local Name Class

LocalName

Definition:	A LocalName references a local object directly accessible from the Name Space.
Subclass Of:	GenericName
Stereotype:	interface

Table B.46 – Member Name Class

MemberName

Definition:	A MemberName is a LocalName that references either an attribute slot in a record or recordType or an attribute, operation, or association role in an object instance or type description in some form of schema.
Subclass Of:	LocalName
Stereotype:	interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
aName	CharacterString [1..1]	[1. The stored value “aName” is the returned value for the “aName()” operation.
attributeType	TypeName [1..1]	The allowable type for this member.

Table B.47 – Namespace Class

NameSpace

Definition:	A Name Space is a domain in which “names” given by character strings (possibly under local constraints) can be mapped to objects via a getObejct operation. Examples include objects which form a Name Space for their attributes, operations and associations, or Schemas that form Name Spaces for their included data types or classes. Not all methods for NameSpaces need to be made publicly accessible.
Stereotype:	interface

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
name	GenericName [0..*]	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
acceptableClassList	TypeName [1..1]	
isGlobal	Boolean [1..1]	

Table B.48 – Scoped Name Class

ScopedName

Definition:	ScopedName is a composite of a LocalName for locating another Name Space and a GenericName valid in that NameSpace. ScopedName contains a LocalName as head and a GenericName, which might be a LocalName or a ScopedName, as tail.
-------------	---

Subclass Of: GenericName

Stereotype: interface

Table B.49 – Type Name Class

TypeName

Definition: A TypeName is a LocalName that references either a recordType or object type in some form of schema. The stored value “aName” is the returned value for the “aName()” operation. This is the types name.

Subclass Of: LocalName

Stereotype: interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
aName	CharacterString [1..1]	1. The stored value “aName” is the returned value for the “aName()” operation.

B.7. Primitive types

The following classes are defined in (ISO 19103:2015)

B.7.1. Date and Time

Table B.50 – Date Class

Date

Definition:

Stereotype: interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
century	CharacterString [1..1]	
day	CharacterString [1..1]	
month	CharacterString [1..1]	
year	CharacterString [1..1]	

Table B.51 – DateTime Class

DateTime

Definition:

Subclass Of: Date and Time

Stereotype: interface

Table B.52 – Time Class

Time

Definition:

Stereotype: interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
hour	CharacterString [1..1]	
minute	CharacterString [1..1]	
second	CharacterString [1..1]	
timeZone	CharacterString [1..1]	

B.7.2. Numerics

Table B.53 – Decimal Class

Decimal

Definition:	The usually finite representation of a decimal number. It differs from the common binary Real implementation in that it can represent 1/10 (one-tenth) without error, while binary real representation can only represent powers of 1/2 (one-half) exactly. Since many currencies are decimal, these representations are preferred in dealing with such moneys. This is also true for mile markers, which are often given in decimals. Can be thought of as an integer part followed by a fractional part given in multiples of powers of 1/10 (tenths).
Subclass Of:	Number
Stereotype:	interface

Table B.54 – Integer Class

Integer

Definition:	An exact integer value, with no fractional part.
Subclass Of:	Number
Stereotype:	interface

Table B.55 – Number Class

Number

Definition:	The base type for all number data, giving the basic algebraic operations. Since all concrete types have finite representations, some part of this algebra for most types exhibit some inaccuracy. For example, Integers cannot divide very well, and reals and decimals cannot avoid certain types of inaccuracies that depend on their representation semantics.
-------------	---

Stereotype: interface

Table B.56 – Real Class

Real

Definition:

The common binary Real finite implementation using base 2. Since such reals can approximate any measure where absolute accuracy is not possible, this form of numeric is most often used for measures. In cases where absolute accuracy is needed, such as currencies, then a decimal representation may be preferred (assuming the currency is decimal, such as the US dollar, British pound, etc.). Where there are no subunits possible, Integer numbers may be preferred. Can be thought of as an integer part followed by a fractional part given in multiples of powers of 1/2 (halves).

Subclass Of: Number

Stereotype: interface

Table B.57 – Unlimited Integer Class

UnlimitedInteger

Definition:

– Infinite if and only if value is not specified {[isInfinte = True] = [value = Null]} – value is either infinite or non-negative {value <> Null implies value ≥ 0 }

Stereotype: interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
-----------	-----------------------------	------------

isInfinite	Boolean [1..1]	
------------	----------------	--

value	Integer [0..1]	
-------	----------------	--

Table B.58 – Vector Class

Vector

Definition:	an ordered set of numbers called coordinates that represent a position in a coordinate system. The coordinates may be in a space of any number of dimensions, as for instance in an “nth degree” polynomial spline. Example (123, 514, 150)
Stereotype:	interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
coordinates	Number [1..*]	list of numbers representing the vector
dimension	Integer [1..1]	dimension in Euclidian space

B.7.3. Text

Table B.59 – Character Set Code Codelist

CharacterSetCode

Definition:
Stereotype: «CodeList»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
ISO10646-1	literal	
ISO10646-2	literal	
ISO8859	literal	

Table B.60 – Character Class

Character

Definition:	symbol from a standard character-set.
-------------	---------------------------------------

Stereotype: interface

Table B.61 – Character String Class

CharacterString

Definition: Characterstring is a family of datatypes which represent strings of symbols from standard character-sets. Semantics of CharacterString is in accordance with ISO/IEC 11404:2007 clause 10.1.5.

Stereotype: interface

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	CharacterString []	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
characterSet	CharacterSetCode [1..1]	character set of the string
maxLength	Integer [1..1]	maximum length of all instances of CharacterString
size	Integer [1..1]	number of characters in the string
value	Character [0..*]	

Table B.62 – URI Class

URI

Definition: Uniform Resource Identifier (URI), is a compact string of characters used to identify or name a resource.

Subclass Of: CharacterString

Stereotype: interface

B.7.4. Truth

Table B.63 – Continuous Truth Class

ContinuousTruth

Definition:	Any of the interpretations of truth as a continuous range of possible values, usually expressed as a measure.
Subclass Of:	Truth
Stereotype:	interface

Table B.64 – Discrete Truth Class

DiscreteTruth

Definition:
Subclass Of:
Stereotype:

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	Boolean []	Realization
	Logical []	Realization

Table B.65 – Probability Class

Probability

Definition:	Continuous truth as a probability. The value (between 0.0 and 1.0) is a measure of belief (in the single occurrence case), or a measure of the distribution of occurrences (in the set case). The fine details of the
-------------	---

semantics is the scope of probability and statistics, and the reader is directed towards his text of choice.
Subclass Of: ContinuousTruth
Stereotype: interface

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
value	Real [1..1]	

Table B.66 – Truth Class

Truth

Definition:	The root or truth classification trees. The only constant between the subclasses here is that 1.0 is TRUE, and 0.0 is FALSE. This makes Probability work nicely. Other algebraic tricks may be more appropriate to other version of “truth calculus”, but for consistency sake, this value mapping should always be available. For example, the following semantic mappings to simple Booleans always work: {truthValue() > 0} = possibly true {truthValue() = 0} = never true {truthValue() < 1} = possibly false {truthValue() = 1} = never false {truthValue() < 1}AND{truthValue() > 0} = uncertain
Stereotype:	interface

Table B.67 – Boolean Enumeration

Boolean

Definition:	boolean is the mathematical datatype associated with two-valued logic
Stereotype:	enumeration

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
		DiscreteTruth []

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
true	literal	one of two possible values of Boolean denoting the primitive value of true
false	literal	one of two possible values of Boolean denoting the primitive value of false

Table B.68 – Logical Enumeration

Logical

Definition:	Handy for a weak sort of 3 valued logic, where truth or falsity is not attributable to all well-formed statements.
Stereotype:	enumeration

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
		DiscreteTruth []

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
TRUE	literal	
FALSE	literal	
MAYBE	literal	The interpretation of MAYBE is a point of semantic variation. One opinion would have it be interpreted as UNKNOWN, implying that certainty exist, but we just don't know what it is at the moment. This is not always appropriate. Even in such a black and white world, a logical when used as a static (class-wide) attribute, would be three valued: always true, always false, and sometimes. Thus, MAYBE can be interpreted as "maybe true, maybe not." In such cases a probability statement might be more appropriate, just not always available. Equivalent to STEP's and SQL's Unknown.

C

ANNEX C (INFORMATIVE) REVISION HISTORY

ANNEX C (INFORMATIVE) REVISION HISTORY

Table C.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2021-06-17	0.0.1	Matthew Purss	all	initial version
2021-07-08	0.0.1	Matthew Purss	Clause 1	initial scope text inserted from original POI draft standard
2021-07-09	0.0.1	Matthew Purss	Clause 4	initial terms inserted from original POI draft standard (and reformatted to meet formal definition requirements)



BIBLIOGRAPHY



BIBLIOGRAPHY

1. Information technology – General-Purpose Datatypes (GPD)
2. Geographic information Observations and measurements
3. Geographic information – Portrayal
4. Geographic information – Services
5. Place holder
6. Place holder
7. Place holder
8. Place holder
9. Place holder
10. Place holder
11. Place holder
12. Place holder
13. Place holder
14. Place holder
15. Place holder
16. Place holder
17. Place holder
18. Place holder
19. Place holder
20. Object Management Group, Model Driven Architecture Guide rev. 2.0