



OGC (POINTS OF INTEREST)

STANDARD

DRAFT

Version: 1.0

Submission Date: 2023-11-20

Approval Date: 2029-03-30

Publication Date: 2023-11-20

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	vii
II.	KEYWORDS	vii
III.	PREFACE	viii
IV.	SECURITY CONSIDERATIONS	ix
V.	SUBMITTING ORGANIZATIONS	x
2.	CONFORMANCE	12
2.1.	Conceptual Models	12
2.2.	Implementation Specifications	12
2.3.	Implementations	13
2.4.	Conformance Classes	13
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
ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)		39
A.1.	Conformance Class A	39
ANNEX B (INFORMATIVE) ISO DATA DICTIONARY		41
B.1.	General Feature Model	41

B.2. Geometry	42
B.3. Citation and responsible party information	45
B.4. Constraint information	48
B.5. Identification information	51
B.6. Name types	53
B.7. Primitive types	56
 ANNEX C (INFORMATIVE) REVISION HISTORY	 66
 BIBLIOGRAPHY	 68

LIST OF TABLES

Table 1	32
Table 2	32
Table 3	33
Table 4	34
Table 5	34
Table 6	34
Table 7	35
Table 8	35
Table 9	36
Table 10	36
Table 11	36
Table 12	37
Table B.1 – Any Feature Class	41
Table B.2 – Feature Type Class	42
Table B.3 – GM_Object Class	43
Table B.4 – GM_Point Class	44
Table B.5 – GM_LineString Class	44
Table B.6 – GM_Polygon Class	45
Table B.7 – CI_Contact Class	45
Table B.8 – CI_Individual Class	46
Table B.9 – CI_Organisation Class	47
Table B.10 – CI_Party Class	47
Table B.11 – CI_Responsibility Class	48
Table B.12	49
Table B.13	49
Table B.14	50
Table B.15	51
Table B.16 – MD_KeywordClass Class	51

Table B.17 – MD_Keywords Class	52
Table B.18 – Generic Name Class	53
Table B.19 – Local Name Class	53
Table B.20 – Member Name Class	54
Table B.21 – Namespace Class	54
Table B.22 – Scoped Name Class	55
Table B.23 – Type Name Class	55
Table B.24 – Date Class	56
Table B.25 – DateTime Class	56
Table B.26 – Time Class	57
Table B.27 – Decimal Class	57
Table B.28 – Integer Class	58
Table B.29 – Number Class	58
Table B.30 – Real Class	58
Table B.31 – Unlimited Integer Class	59
Table B.32 – Vector Class	59
Table B.33 – Character Set Code Codelist	59
Table B.34 – Character Class	60
Table B.35 – Character String Class	60
Table B.36 – URI Class	61
Table B.37 – Continuous Truth Class	61
Table B.38 – Discrete Truth Class	61
Table B.39 – Probability Class	62
Table B.40 – Truth Class	62
Table B.41 – Boolean Enumeration	63
Table B.42 – Logical Enumeration	63
Table C.1	66

LIST OF FIGURES

Figure 1 – UML notation (see ISO TS 19103, Geographic information - Conceptual schema language).	20
Figure 2 – Example UML diagram demonstrating the UML notation and coloring scheme used throughout the POI Standard.	22
Figure 3 – Feature Model	25
Figure 4 – Geometry Model	27
Figure 5 – POI UML Model - ISO Extensions	29
Figure 6 – POI UML Model - Core	30
Figure 7 – 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 A.1: /req/req-class-a/req-name-1	39

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 optional POI properties

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.

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. Implementations

POI implementations will typically be a simplified representation of a more complex dataset. Implementors may want to extend the POI model to include properties specific to that dataset. These extensions are accomplished using the POI Payload mechanism described in POI Payload. Since the POI Payload has its own definition of syntax and semantics, conformance with the POI Standard cannot ensure payload conformance.

2.4. 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 this 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.

ISO: ISO 19101-1:2014, *Geographic information – Reference model – Part 1: Fundamentals*, 2014, <https://www.iso.org/standard/59164.html>

ISO: ISO 19103:2015, *Geographic information – Conceptual schema language*, 2015, <https://www.iso.org/standard/56734.html>

ISO: ISO 19107:2003, *Geographic Information – Spatial Schema*, 2003, <https://www.iso.org/standard/26012.html>

ISO: ISO 19109:2015, *Geographic Information – Rules for Application Schemas*, 2015, <https://www.iso.org/standard/59193.html>

ISO: ISO 19115-1:2014, *Geographic information – Metadata – Part 1: Fundamentals*, 2014, <https://www.iso.org/standard/53798.html>

ISO/IEC 19507:2012, *Information technology – Object Management Group Object Constraint Language (OCL)*, 2012, <https://www.iso.org/standard/57306.html>

OGC: The Specification Model – A Standard for Modular specifications, OGC 08-131r3, 10/29/2009, https://portal.ogc.org/files/?artifact_id=34762&version=2

OMG: Object Management Group (OMG) *Unified Modeling Language (UML)*, Version 2.5.1, December 2017, <https://www.omg.org/spec/UML/2.5.1>

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

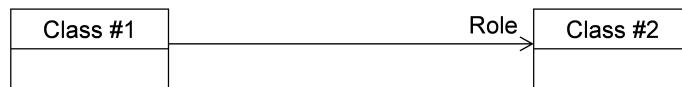
<http://www.opengis.net/spec/POI/1.0>

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

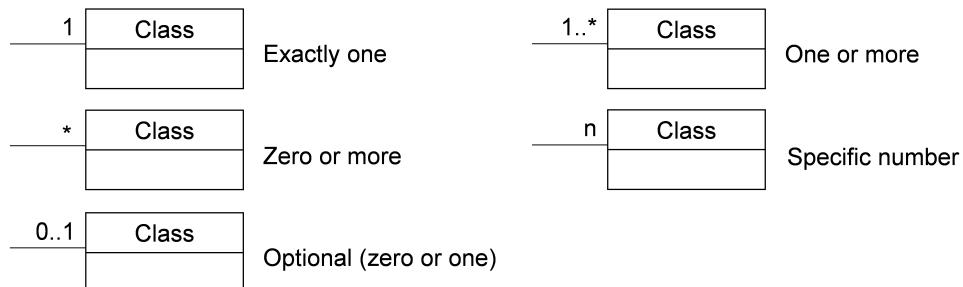
5.2. UML Notation

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 1.

Association between classes



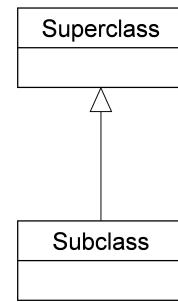
Association cardinality



Aggregation between classes



Class inheritance



Composition between classes



Figure 1 – 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

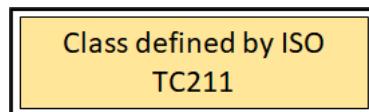
POI Conceptual Model only for classes that are imported from the ISO standards 19103, 19107, 19109, and 19115.

- «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.

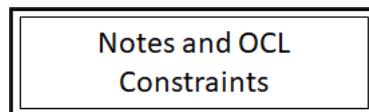
In order to enhance the readability of the POI UML diagrams, classes are depicted in different colors. The following coloring scheme is applied:



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 2 demonstrates the UML notation and coloring scheme used throughout this standard. In this example, the yellow classes are associated with the ISO

Standards, the green classes are from the *POI Example Classes*, and the white classes depicts notes.

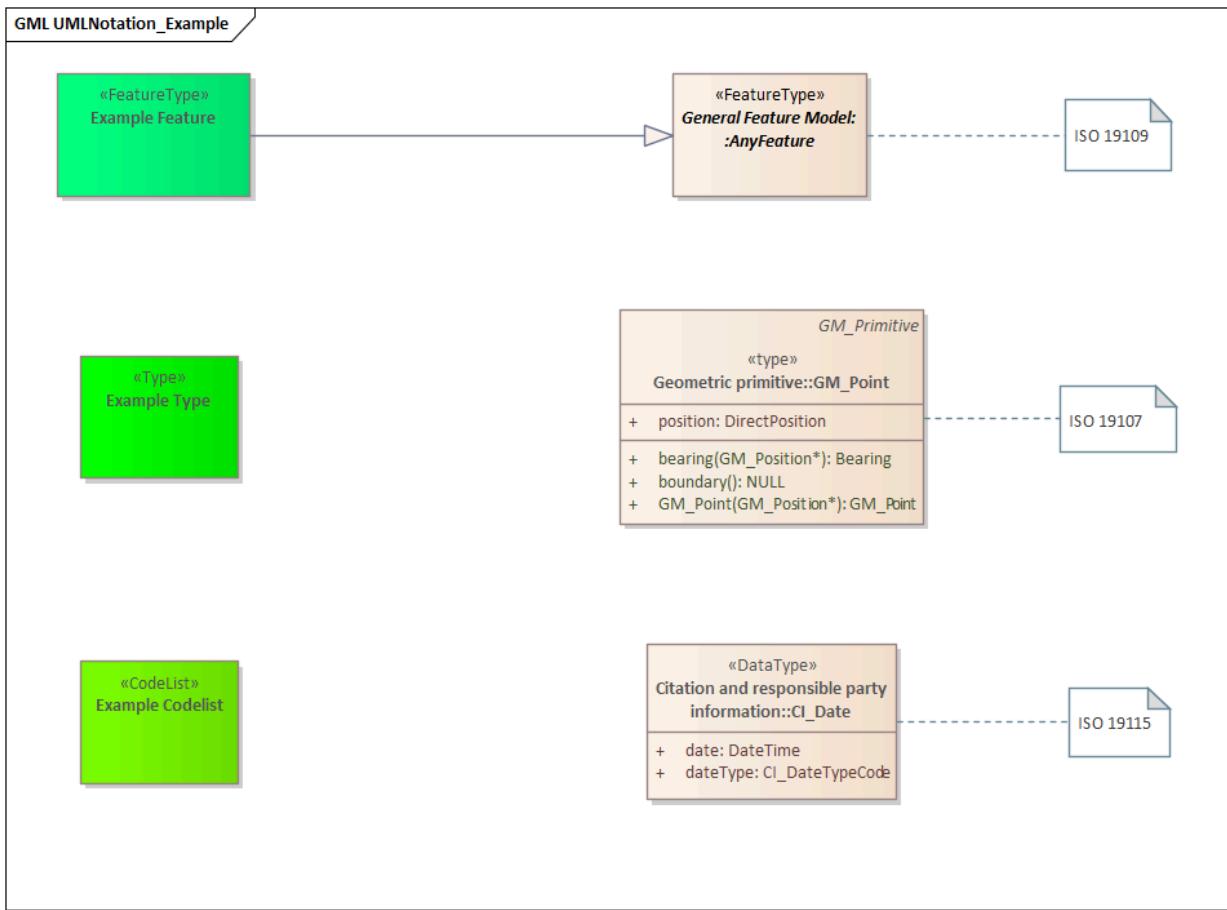


Figure 2 – Example UML diagram demonstrating the UML notation and coloring scheme used throughout the POI Standard.

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 3.

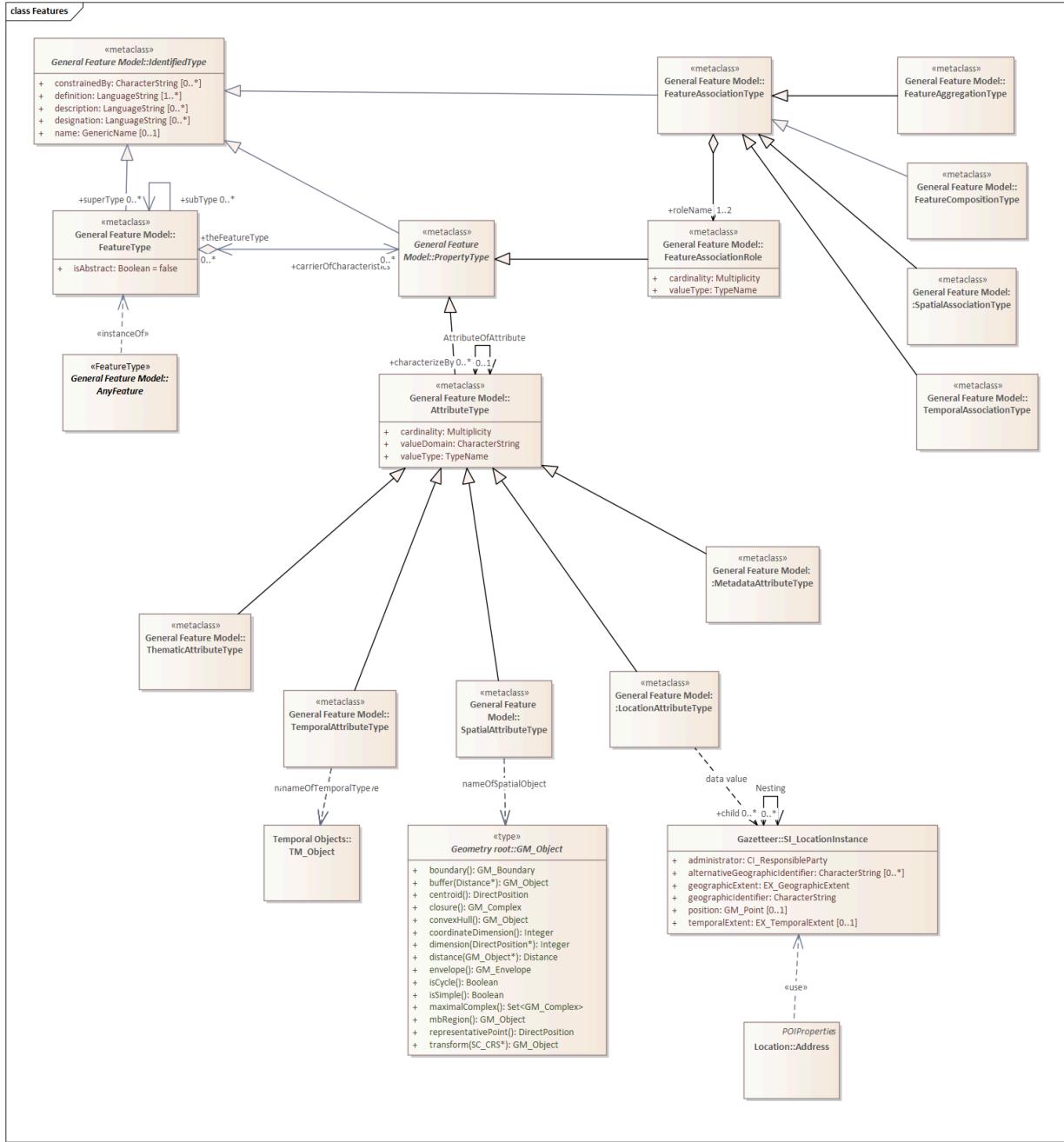


Figure 3 – Feature Model

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. Therefore, 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 4 provides a UML model of the classes from ISO 19107 which are applicable to POIs.

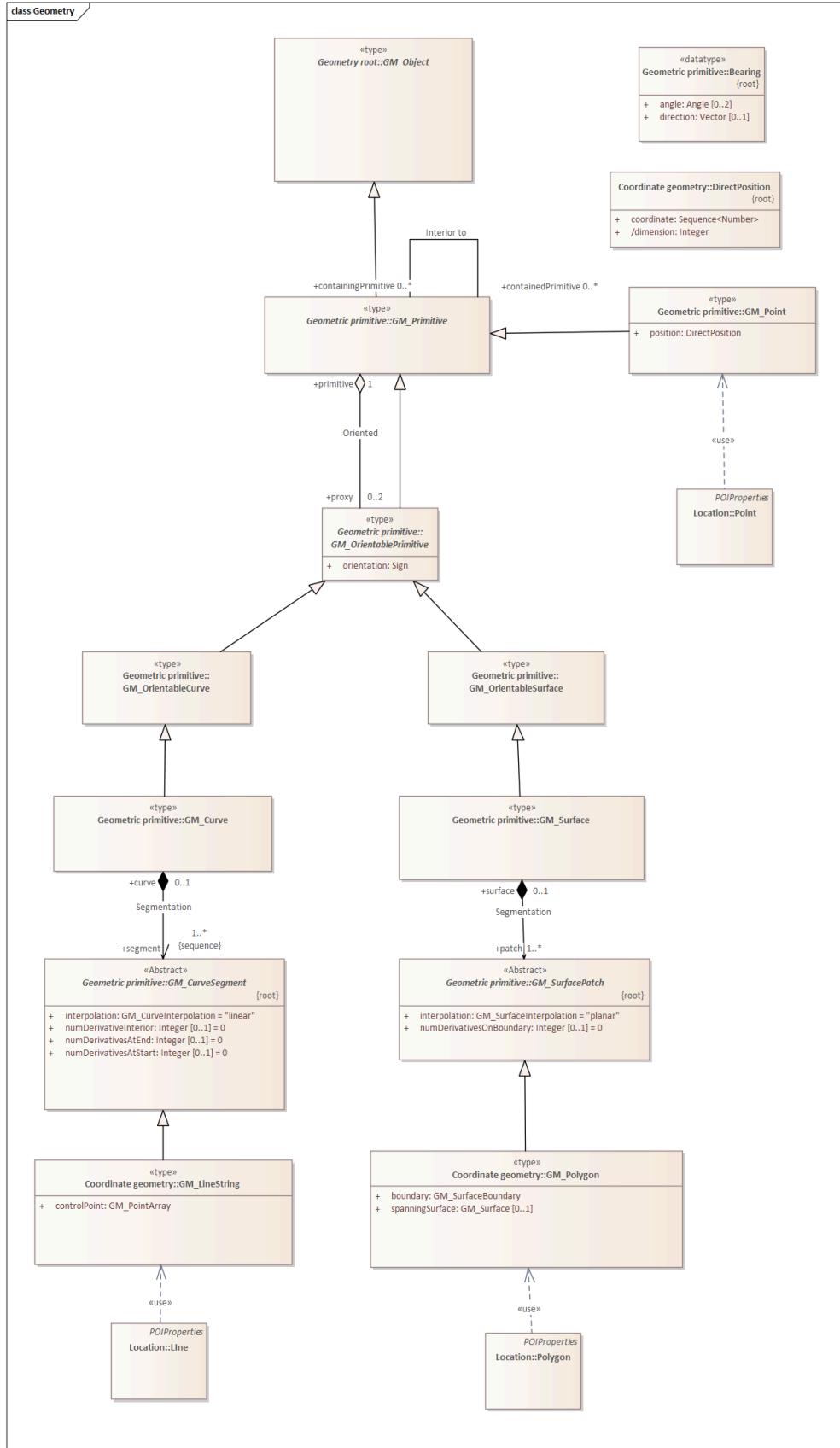


Figure 4 – Geometry Model

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.

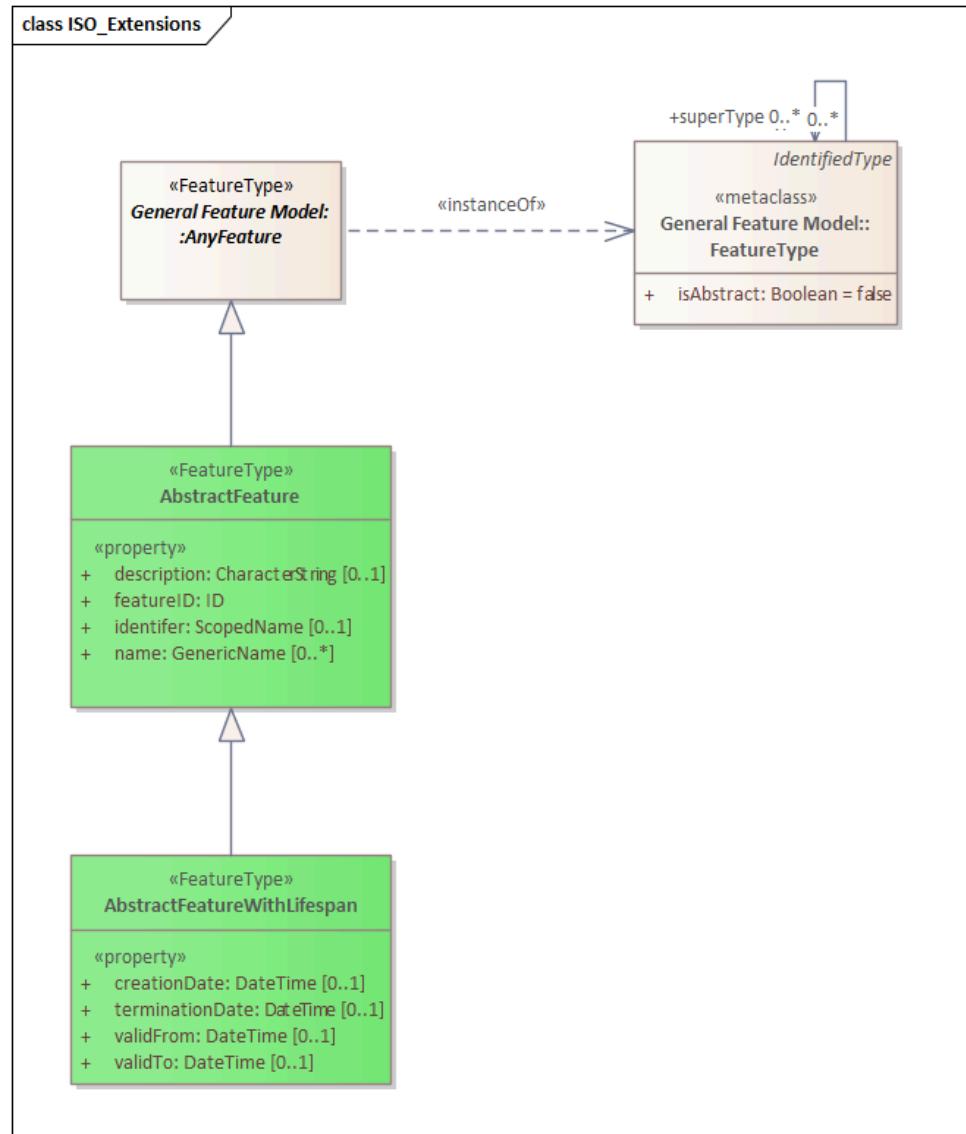


Figure 5 – POI UML Model - ISO Extensions

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. *AbstractFeature* adds descriptive and identifying properties to *AnyFeature*.

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.

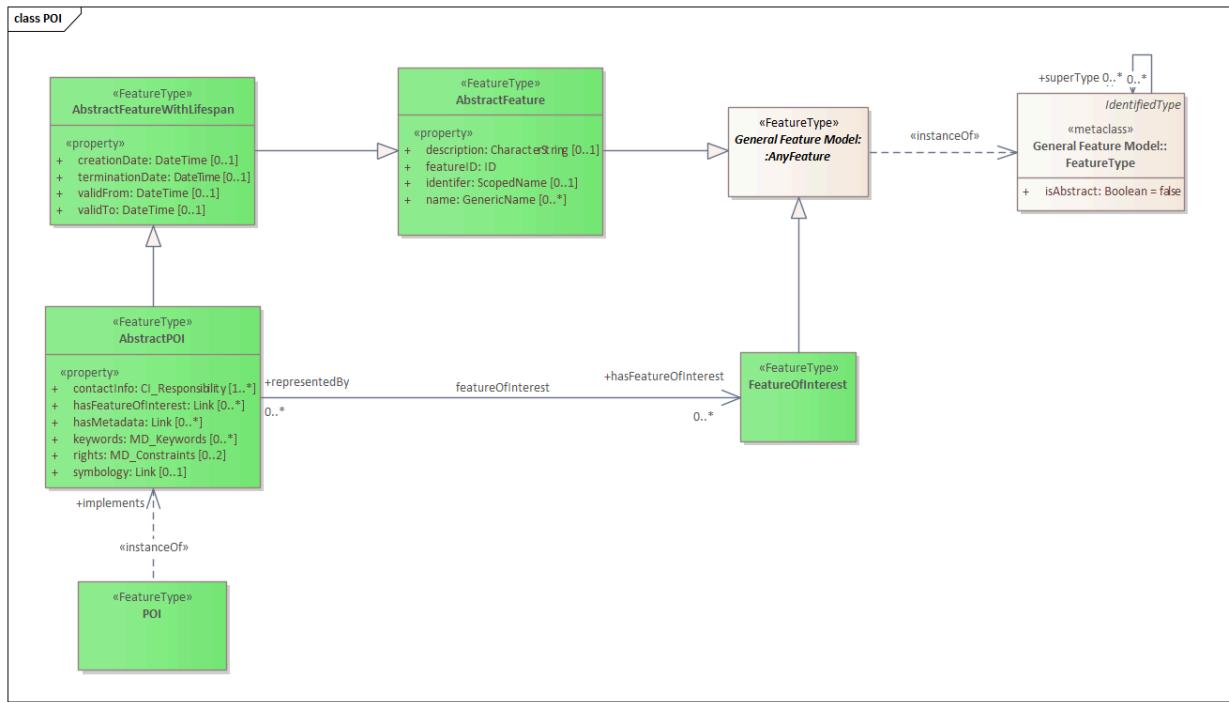


Figure 6 – POI UML Model - Core

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.

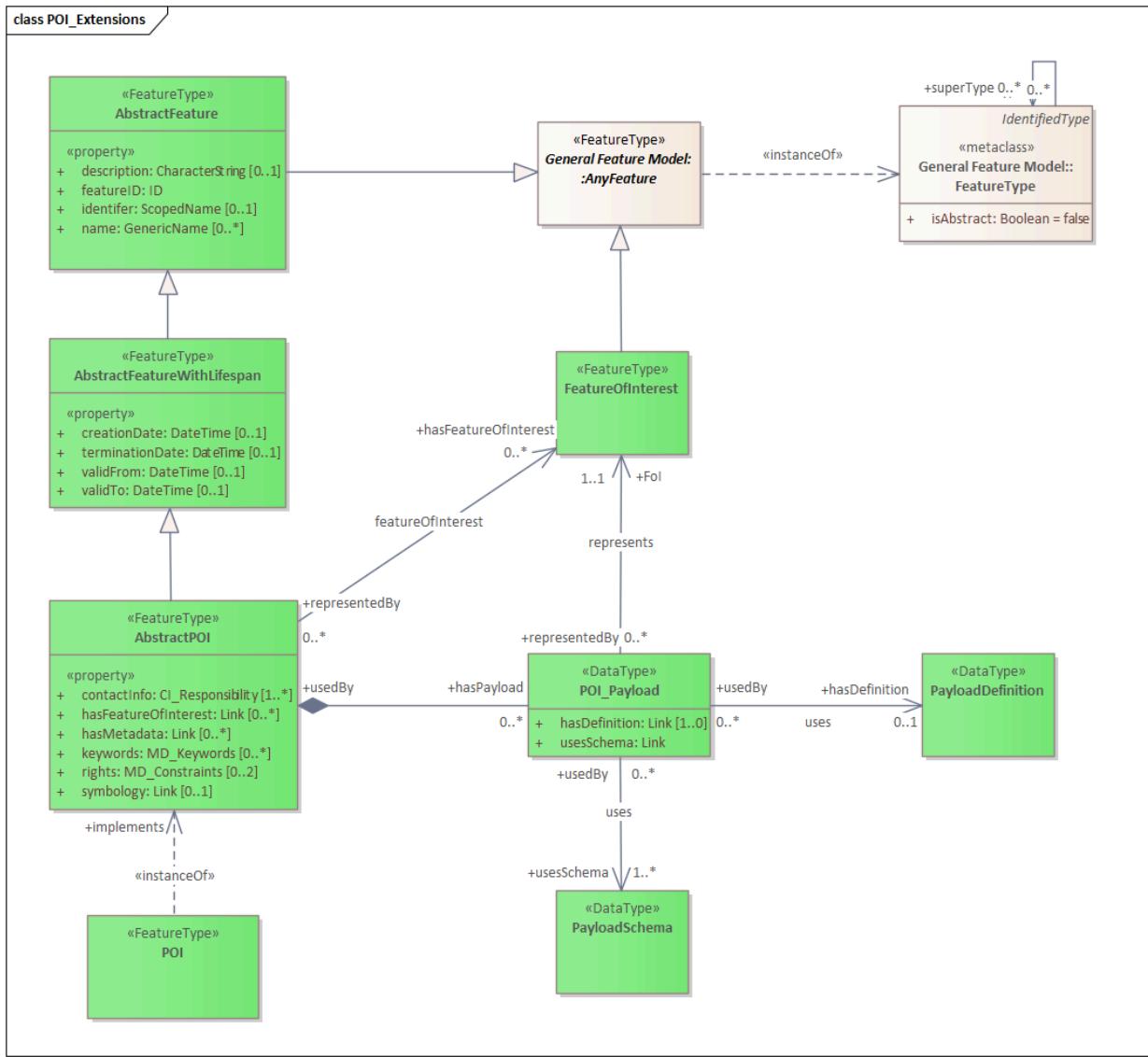


Figure 7 – POI UML Model - Payload

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.

In the interest of interoperability, the POI Payload should be constructed using data types and concepts which are already in wide use by the Geospatial community. A collection of data types

and concepts defined by the ISO 19103, ISO 19107, ISO 19109, and ISO 19115 Standards is provided in Annex B.

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

AbstractFeature

Definition:	AbstractFeature is the abstract superclass of all feature types within the PoI 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	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.

Table 2

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		
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.

Table 3

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
hasFeature OfInterest	FeatureOfInterest [0..*]	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.
hasFeature OfInterest «property»	Link [0..*]	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.

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
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.
symbology «property»	Link [0..1]	A reference to information about rendering this POI.

Table 4

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»

Table 5

PayloadDefinition

Definition:	The semantic model (ontology) for a POI payload.
Subclass of:	None
Stereotype:	«DataType»

Table 6

PayloadSchema

Definition:	A model of the syntax of the POI payload.
Subclass of:	None

Stereotype:	«DataType»
-------------	------------

Table 7

POI

Definition:	An instance of a POI. Implements the AbstractPOI class.
-------------	---

Subclass of:	None
--------------	------

Stereotype:	«FeatureType»
-------------	---------------

Table 8

POI_Payload

Definition:	A representation of properties of the Fol which are to be included in the POI.
-------------	--

Subclass of:	None
--------------	------

Stereotype:	«DataType»
-------------	------------

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
usedBy	AbstractPOI []	
hasDefinition	PayloadDefinition [0..1]	
usesSchema	PayloadSchema [1..*]	
Fol	FeatureOfInterest [1..1]	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
hasDefinition	Link [1..0]	A reference to the semantic model of the payload for this POI. The target of this reference is an instance of the PayloadDefinition class.
usesSchema	Link	A reference to the schema for the payload of this POI. The target of this reference is an instance of the PayloadSchema class.

Table 9

ID

Definition:	ID is a basic type that represents a unique identifier.
Subclass of:	None
Stereotype:	«Type»

Table 10

Integer

Definition:	
Subclass of:	None
Stereotype:	«Type»

Table 11

Link

Definition:	
Subclass of:	None
Stereotype:	«Type»

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
href	String	Supplies the URI to a remote resource (or resource fragment).
hreflang	String [0..1]	A hint indicating what the language of the result of dereferencing the link should be.
length	int	

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
rel	String	The type or semantics of the relation.
title	String [0..1]	Used to label the destination of a link such that it can be used as a human-readable identifier.
type	String [0..1]	A hint indicating what the media type of the result of dereferencing the link should be.

Table 12

String

Definition:	
Subclass of:	None
Stereotype:	«Type»



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 – 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.3 – 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.4 – 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.5 – 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.6 – 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.7 – 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
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.8 – 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.9 – 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_BrowseGraphic [0..*]	Graphic identifying organization

Table B.10 – 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.11 – 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

B.4. Constraint information

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

Table B.12

MD_Constraints

ROLE NAME	TARGET CLASS AND MULTIPLICITY	DEFINITION
	MD_Identification []	
	MD_Metadata []	
ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
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.13

MD_LegalConstraints

Definition:	restrictions and legal prerequisites for accessing and using the resource or metadata
-------------	---

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

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.14

MD_Releasability

Definition:	information about resource release constraints
Subclass Of:	None
Stereotype:	
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.15

MD_SecurityConstraints

ATTRIBUTE	VALUE TYPE AND MULTIPLICITY	DEFINITION
classification	MD_Classification Code	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.16 – 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.17 – 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_KeywordType Code [0..1]	subject matter used to group similar keywords

B.6. Name types

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

Table B.18 – 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.19 – Local Name Class

LocalName

Definition:	A LocalName references a local object directly accessible from the Name Space.
-------------	--

Subclass Of: GenericName

Stereotype: interface

Table B.20 – 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]	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.21 – Namespace Class

NameSpace

Definition: A Name Space is a domain in which “names” given by character strings (possibly under local constraints constraints enforced by the Name Space) can be mapped to objects via a getObejt 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
acceptable ClassList	TypeName [1..1]	
isGlobal	Boolean [1..1]	

Table B.22 – 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.23 – 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]	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.24 – 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.25 – DateTime Class

DateTime

Definition:

Subclass Of: Date and Time

Stereotype: interface

Table B.26 – 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.27 – 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.28 – Integer Class

Integer

Definition: An exact integer value, with no fractional part.

Subclass Of: Number

Stereotype: interface

Table B.29 – 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.30 – 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.31 – 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.32 – 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.33 – Character Set Code Codelist****CharacterSetCode**

Definition:

Stereotype: «CodeList»

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

Table B.34 – Character Class

Character

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

Table B.35 – 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.36 – 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.37 – 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.38 – Discrete Truth Class

DiscreteTruth

Definition:
Subclass Of: Truth
StereoType: interface

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

Table B.39 – 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.40 – 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

	<p>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</p>
Stereotype:	interface

Table B.41 – 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.42 – 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	<p>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.</p>

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. Place holder
2. Information technology — General-Purpose Datatypes (GPD)
3. Geographic information Observations and measurements
4. Geographic information — Portrayal
5. Geographic information — Services
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