



BSI Standards Publication

Geographic information — General feature model and rules for application schema

National foreword

This British Standard is the UK implementation of EN ISO 19109:2025. It is identical to ISO 19109:2025. It supersedes BS EN ISO 19109:2015, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/36, Geographic information.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 21459 8

ICS 35.240.70

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2025.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Geographic information - General feature model and rules for application schema (ISO 19109:2025)

Information géographique - Modèle général des entités
et règles relatives au schéma d'application (ISO
19109:2025)

Geoinformation - Grundlegendes Datenmodell und
Regeln zur Erstellung von Anwendungsschemata (ISO
19109:2025)

This European Standard was approved by CEN on 20 June 2025.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

This document (EN ISO 19109:2025) has been prepared by Technical Committee ISO/TC 211 "Geographic information/Geomatics " in collaboration with Technical Committee CEN/TC 287 "Geographic Information" the secretariat of which is held by BSI.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by January 2026, and conflicting national standards shall be withdrawn at the latest by January 2026.

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

This document supersedes EN ISO 19109:2015.

Any feedback and questions on this document should be directed to the users' national standards body/national committee. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Endorsement notice

The text of ISO 19109:2025 has been approved by CEN as EN ISO 19109:2025 without any modification.

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions, and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	4
4 Conformance	4
4.1 General	4
4.2 Meta-Model	4
4.3 Spatial	4
4.4 Temporal	4
4.5 Quality	5
4.6 Spatial referencing by identifiers	5
4.7 Coverages	5
4.8 UML application schema	5
4.9 Profile existing conceptual schema	5
5 Presentation and abbreviations	6
5.1 Presentation	6
5.1.1 General	6
5.1.2 Conformance class	6
5.1.3 Requirements class	6
5.1.4 Provisions	6
5.1.5 Identifiers	7
5.1.6 Conceptual schemas	7
5.1.7 Descriptions of concepts	7
5.2 Package abbreviations	7
6 Context	8
6.1 Purpose of an application schema	8
6.2 Rationale for defining the rules for application schemas	8
6.3 Application schemas supporting data interchange	9
7 General feature model	10
7.1 Principle for defining features	10
7.1.1 Features, coverages and properties	10
7.1.2 Essential properties of features	11
7.2 The Concept of the General Feature Model	12
7.2.1 Introduction	12
7.2.2 The purpose of the GFM	12
7.3 Conceptual Schema of the General Feature Model	13
7.3.1 The structure of the GFM	13
7.3.2 The main part of the GFM	14
7.3.3 Metaclass IdentifiedType	15
7.3.4 Metaclass FeatureType	16
7.3.5 Metaclass PropertyType	16
7.3.6 Metaclass AttributeType	16
7.3.7 Metaclass FeatureAssociationRole	17
7.3.8 Metaclass ValueAssignment	17
7.3.9 Metaclass FeatureAssociationType	18
7.3.10 Metaclass InheritanceRelation	18
7.4 Attributes of feature types	18
7.4.1 Introduction	18
7.4.2 Metaclass SpatialAttributeType	19

7.4.3	Metaclass TemporalAttributeType.....	19
7.4.4	Metaclass QualityAttributeType.....	19
7.4.5	Metaclass LocationAttributeType.....	19
7.4.6	Metaclass MetadataAttributeType.....	20
7.4.7	Metaclass ThematicAttributeType.....	20
7.4.8	Metaclass CoverageFunctionAttributeType.....	20
7.5	Relationships between feature types.....	20
7.5.1	Introduction.....	20
7.5.2	Metaclass InheritanceRelation.....	20
7.5.3	Metaclass FeatureAssociationType.....	21
7.6	Constraints.....	21
8	Rules for application schemas.....	22
8.1	The application modelling process.....	22
8.1.1	Introduction.....	22
8.1.2	Features and the application schema.....	22
8.2	General rules for application schemas.....	23
8.2.1	Rule for using CSLs.....	24
8.2.2	Rule for integration.....	24
8.2.3	Rules for modelling features.....	24
8.2.4	Rule for property names.....	25
8.2.5	Rule for attributes.....	25
8.2.6	Rule for association roles.....	25
8.2.7	Rule for value assignments.....	26
8.2.8	Rule for feature associations.....	26
8.2.9	Rule for inheritance relations.....	26
8.3	Rules for use of spatial schemas.....	26
8.3.1	Rules for modelling applications with spatial properties.....	26
8.3.2	Use of standard spatial schemas.....	26
8.3.3	Rule for spatial attributes.....	27
8.3.4	Use of geometric collections and spatial complexes to represent the values of spatial attributes of features.....	28
8.3.5	Spatial associations between features.....	28
8.3.6	Features sharing geometry.....	29
8.3.7	Point features, line features and area features.....	30
8.3.8	Defining interpolation methods.....	30
8.3.9	Independent spatial complexes.....	31
8.4	Rules for use of temporal schemas.....	33
8.4.1	Rules for modelling applications with temporal properties.....	33
8.4.2	Use of temporal conceptual schema.....	34
8.4.3	Rule for temporal attributes.....	34
8.4.4	Temporal associations between features.....	34
8.5	Rules for use of quality schemas.....	35
8.5.1	Introduction.....	35
8.5.2	Data quality rules.....	35
8.6	Rule for use of geographic identifiers.....	37
8.7	Use of metadata.....	39
8.8	Rule for use of coverage functions.....	39
8.9	Use of observations.....	41
8.10	Rules for application schemas in UML.....	44
8.10.1	General.....	44
8.10.2	Rules for conceptual schema language for application schemas.....	44
8.10.3	Rule for packaging and identification of an application schema.....	47
8.10.4	Documentation of an application schema.....	48
8.10.5	Rules for integration of application schemas and abstract schemas.....	48
8.10.6	Rules for modelling structures in UML.....	50
8.10.7	Linguistic adaptation.....	52
8.11	Rules for domain profiles of existing conceptual schemas in UML.....	53
8.11.1	Introduction.....	53

ISO 19109:2025(en)

8.11.2	Rule for adding information to an existing conceptual schema.....	54
8.11.3	Rule for tailored use of an existing conceptual schema.....	55
Annex A	(normative) Abstract test suite.....	57
Annex B	(informative) The modelling approach and the General Feature Model.....	63
Annex C	(informative) Application schema examples	66
Annex D	(informative) Backward compatibility	71
Bibliography		76

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 287, *Geographic Information*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

This third edition cancels and replaces the second edition (ISO 19109:2015), which has been technically revised.

The main changes are as follows:

- changes in the title and scope;
- new sub-clauses discussing the concept of the General Feature Model;
- re-organization of [Clause 7](#) to include only concepts of the General Feature Model and moving the general rules for application schema to [Clause 8](#);
- updating the references to other ISO/TC 211 standards in applicable cases to reflect classes in respective latest versions;
- removing the dependencies to other ISO/TC 211 standards related to attributes of features.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Any description of reality is always an abstraction, always partial, and always just one of many possible “views”, depending on the application field.

The widespread application of computers and geographic information systems (GIS) has led to an increased use of geographic data within multiple disciplines. With current technology as an enabler, society's reliance on such data is growing. Geographic datasets are increasingly being shared and exchanged. They are also used for purposes other than those for which they were produced.

To ensure that data will be understood by both computer systems and users, it is necessary to fully document the data structures for data access and exchange. The interfaces between systems, therefore, need to be defined with respect to data and operations, using the methods standardized in this document. For the construction of internal software and data storage within proprietary systems, any method is acceptable provided it supports the standardized interfaces.

An application schema provides the formal description of the data structure and content required by one or more applications. An application schema contains the descriptions of both geographic data and other related data. A fundamental concept of geographic data is the feature.

This document aims to express the importance of continuing the modelling of geospatial information according to the concepts contained in this document. The name and contact information of the maintenance agency for this document can be found at www.iso.org/maintenance_agencies.

Geographic information — General feature model and rules for application schema

1 Scope

This document defines the General Feature Model (GFM) as the metamodel for creating application schemas in the context of geo-information modelling. The GFM is explained and implemented as rules for creating and documenting application schemas, including principles for the definition of features.

This document is applicable to:

- conceptual modelling of features and their properties from a universe of discourse;
- definition of application schemas;
- general rules for using a conceptual schema language for application schemas;
- rules for application schemas using UML as the conceptual schema language;
- transition from the concepts in the conceptual model to the data types in the application schema;
- integration of standardized schemas from other ISO geographic information standards with the application schema.

This document does not apply to:

- choice of one particular conceptual schema language for application schemas;
- definition of any particular application schemas;
- representation of feature types and their properties in a feature catalogue;
- representation of metadata;
- rules for mapping one application schema to another;
- implementation of the application schema in a computer environment;
- computer system and application software design;
- programming.

2 Normative references

There are no normative references in this document.

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>

— IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1

application

manipulation and processing of data in support of user requirements

[SOURCE: ISO 19101-1:2014, 4.1.1]

3.1.2

application schema

conceptual schema ([3.1.5](#)) for data required by one or more *applications* ([3.1.1](#))

[SOURCE: ISO 19101-1:2014, 4.1.2]

3.1.3

complex feature

feature ([3.1.9](#)) composed of other features

3.1.4

conceptual model

model ([3.1.15](#)) that defines concepts of a *universe of discourse* ([3.1.19](#))

[SOURCE: ISO 19101-1:2014, 4.1.5]

3.1.5

conceptual schema

formal description of a *conceptual model* ([3.1.4](#))

[SOURCE: ISO 19101-1:2014, 4.1.6]

3.1.6

coverage

function which returns values from its range for any direct position within its *domain* ([3.1.8](#))

[SOURCE: ISO 19123-1:2023, 3.1.9]

3.1.7

dataset

identifiable collection of data

[SOURCE: ISO 19115-1:2014, 4.3]

3.1.8

domain

well-defined set

Note 1 to entry: Well-defined means that the definition is both necessary and sufficient, as everything that satisfies the definition is in the set and everything that does not satisfy the definition is necessarily outside the set.

3.1.9

feature

abstraction of real-world phenomena

Note 1 to entry: A feature can occur as a type or an instance. Feature type or feature instance will be used when only one is meant.

[SOURCE: ISO 19101-1:2014, 4.1.11]

3.1.10

feature association

relationship that links instances of one *feature* ([3.1.9](#)) type with instances of the same or a different feature type

[SOURCE: ISO 19110:2016, 3.3]

3.1.11

feature attribute

characteristic of a *feature* ([3.1.9](#))

Note 1 to entry: A feature attribute can occur as a type or an instance. Feature attribute type or feature attribute instance will be used when only one is meant.

Note 2 to entry: A feature attribute type has a name, a data type and a domain associated with it. A feature attribute instance has an attribute value taken from the domain of the feature attribute type.

[SOURCE: ISO 19101-1:2014, 4.1.12, modified — EXAMPLES and Notes have been removed and two new Notes to entry have been added.]

3.1.12

geographic data

data with implicit or explicit reference to a location relative to the Earth

Note 1 to entry: Geographic information is also used as a term for information concerning phenomena implicitly or explicitly associated with a location relative to the Earth.

3.1.13

general feature model

metamodel that classifies real-world phenomena and defines their respective attributes and relationships

3.1.14

metadata

information about a resource

[SOURCE: ISO 19115-1:2014, 4.10]

3.1.15

model

abstraction of some aspects of reality

3.1.16

observation

act carried out by an observer to determine the value of an observable *property* ([3.1.17](#)) of an object (feature-of-interest) by using a procedure, with the value provided as the result

[SOURCE: ISO 19156:2023, 3.13]

3.1.17

property

facet or attribute of an object referenced by a name

[SOURCE: ISO 19143:2010, 4.21]

3.1.18

quality

degree to which a set of inherent characteristics of an object fulfils requirements

[SOURCE: ISO 9000:2015, 3.6.2, modified — Note 1 and Note 2 to entry have been removed.]

3.1.19

universe of discourse

view of the real or hypothetical world that includes everything of interest

[SOURCE: ISO 19101-1:2014, 4.1.38]

3.2 Abbreviated terms

CSL	Conceptual Schema Language
GFM	General Feature Model
MOF	Meta-Object Facility
OCL	Object Constraint Language
OWL	Web Ontology Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier

4 Conformance

4.1 General

This document defines 8 conformance classes shown in [Tables 1](#) to [8](#), matching the 8 requirements classes described in [Clause 8](#). Any application schema claiming conformance to any requirements class in this document shall pass all of the tests listed in the corresponding conformance class, which are described in detail in the abstract test suites in [Annex A](#). Each test relates to one or more specific requirements, which are explicitly indicated in the description of the test.

4.2 Meta-Model

Table 1 — Meta-model conformance class

Conformance class	/conf/general
Requirements class	/req/general (8.2 , Table 11)
Tests	All tests in Clause A.2

4.3 Spatial

Table 2 — Spatial conformance class

Conformance class	/conf/general-spatial
Dependency	/conf/general (4.2)
Requirements class	/req/general-spatial (8.3 , Table 12)
Tests	All tests in Clause A.3

4.4 Temporal

Table 3 — Temporal conformance class

Conformance class	/conf/general-temporal
Dependency	/conf/general (4.2)
Requirements class	/req/general-temporal (8.4 , Table 14)
Tests	All tests in Clause A.4

4.5 Quality

Table 4 — Quality conformance class

Conformance class	/conf/general-quality
Dependency	/conf/general (4.2)
Requirements class	/req/general-quality (8.5 , Table 16)
Tests	All tests in Clause A.5

4.6 Spatial referencing by identifiers

Table 5 — Spatial referencing by identifiers conformance class

Conformance class	/conf/general-identifier
Dependency	/conf/general (4.2)
Requirements class	/req/general-identifier (8.6 , Table 17)
Tests	All tests in Clause A.6

4.7 Coverages

Table 6 — Coverages conformance class

Conformance class	/conf/general-coverage
Dependency	/conf/general (4.2)
Requirements class	/req/general-coverage (8.8 , Table 18)
Tests	All tests in Clause A.7

4.8 UML application schema

Table 7 — UML application schema conformance class

Conformance class	/conf/uml
Dependency	/conf/general (4.2)
Requirements class	/req/uml (8.10.1 , Table 20)
Tests	All tests in Clause A.8

4.9 Profile existing conceptual schema

Table 8 — Profile existing conceptual schema conformance class

Conformance class	/conf/uml-profile
Dependency	/conf/uml (4.8)
Requirements class	/req/uml-profile (8.11 , Table 22)
Tests	All tests in Clause A.9

5 Presentation and abbreviations

5.1 Presentation

5.1.1 General

This document describes how to create an application schema that integrates conceptual schemas defined in the ISO geographic information standards. In addition to stating the rules for creating application schemas, this document provides guidance through examples.

5.1.2 Conformance class

Conformance to this document is possible at a number of levels, specified by conformance classes ([Clause 4](#)). Each conformance class is summarized using the template shown as [Table 9](#).

Table 9 — Conformance class template

Conformance class	/conf/{classM}
Dependency	[identifier for another conformance class]
Requirements class	/req/{classA}
Tests	[reference to clause(s) containing tests]

All tests in a class shall be passed, so dependencies are recorded with respect to other conformance classes rather than individual tests. Each conformance class tests conformance to a set of requirements packaged in a requirements class ([Clause 8](#)).

5.1.3 Requirements class

Each requirement or recommendation in this document is a member of a requirements class. In this document each requirements class is described in a discrete clause or subclause and summarized using the template shown as [Table 10](#).

Table 10 — Requirements class template

Requirements class	/req/{classM}
Target type	[artefact or technology type]
Dependency	[identifier for another requirements class]
Requirement	/req/{classM}/{reqN}
Recommendation	/rec/{classM}/{recO}
Requirement/Recommendation	[repeat as necessary]

All requirements in a class shall be satisfied, so the requirements class is the unit of re-use and dependency, rather than individual requirements. Hence, the value of a dependency requirement is another requirements class.

5.1.4 Provisions

All requirements are normative, while recommendations convey a possible course of action deemed suitable without necessarily prohibiting others. Each provision is presented using the following template:

Name # /re(c|q)/[classM]/[re(c|q)(N|O)]

[Normative statement]

where /req/[classM]/[reqN] identifies the requirement and /rec/[classM]/[recO] identifies the recommendation. The use of this layout convention allows the provisions of this document to be easily located by implementers.

5.1.5 Identifiers

Each requirements class, requirement, and recommendation has an identifier in the form of a path or partial URI. The identifier supports cross-referencing of class membership, dependencies, and links from each conformance test to requirements tested. The identifier can be appended to a URI that identifies the standard as a whole in order to construct an absolute URI which identifies the requirements class, requirement or recommendation. For example, a URI scheme following the approach described in 4.6 of IETF RFC 5141^[27] would result in:

— <https://standards.isotc211.org/19109/-/3>

Hence, the absolute URI for each requirements class would have the form

— [https://standards.isotc211.org/19109/-/3/req/\[classM\]](https://standards.isotc211.org/19109/-/3/req/[classM])

and the absolute URI for each requirement or recommendation would have the form

— [https://standards.isotc211.org/19109/-/3/re\(c\[q\]\)/\[classM\]/\[re\(c\[q\]\)\(N|O\)\]](https://standards.isotc211.org/19109/-/3/re(c[q])/[classM]/[re(c[q])(N|O)])

5.1.6 Conceptual schemas

Conceptual schemas in the normative part of this document are presented in the Unified Modeling Language (UML) in conformance with ISO 19103. UML diagrams are presented in conformance with UML 2.5.1.

5.1.7 Descriptions of concepts

Within the diagrams in this document, as a convention, class names are denoted as in Recommendation 12 of ISO 19103.

Within the text, class names and attributes are presented in italics for clarity of discussion (e.g., *AttributeType*). When necessary, such as in cases where two interfaces from different packages have the same name, the package name is written before the name, separated by double colons (e.g., *Topology::Complex* vs. *Geometry::Complex*),

Concepts from UML are presented in all capitals – e.g., CLASS, PACKAGE, ROLE, ATTRIBUTE, ASSOCIATION.

5.2 Package abbreviations

Concepts from schemas defined in some other International Standards are designated with names that start with two-letter codes as follows:

EX	ISO 19115-1:2014/Amd 1:2018/Amd 2:2020 Extent Information
LI	ISO 19115-1:2014/Amd 1:2018/Amd 2:2020 Lineage information
MD	ISO 19115-1:2014/Amd 1:2018/Amd 2:2020 Metadata fundamentals
MI	ISO 19115-2:2019/Amd 1:2022 Metadata extensions for imagery
TM	ISO 19108:2002/Cor 1:2006 Temporal Schema

6 Context

6.1 Purpose of an application schema

An application schema is a conceptual schema for data required by one or more applications. An application schema defines:

- the content and structure of data;

and can define:

- operations for manipulating and processing data;
- constraints to ensure the integrity of the data.

The purpose of an application schema is twofold:

- to provide a computer-readable data description defining the data structure, which makes it possible to apply automated mechanisms for data management;
- achieve a common and correct understanding of the data, by documenting the data content of the particular application field, thereby making it possible to unambiguously retrieve information from the data.

6.2 Rationale for defining the rules for application schemas

This document does not standardize any application schemas; it only defines rules for creating application schemas in a consistent manner (including the consistent definition of features) to facilitate the acquiring, processing, analysing, accessing, presenting and transferring of geographic data between different users, systems and locations. The rules in this document are, in the case of data transfer or interchange, used by suppliers and users of geographic data to:

- build a transfer application schema for data interchange;
- interpret the semantics of the transferred dataset with respect to user's local data and content and structure of data;
- determine the necessary transformations between the two datasets.

The rules in this document will assist the users of applications with similar data requirements in creating a common application schema for the interface between their systems and data. This includes an agreement about the elements from the universe of discourse. This is described in more detail in [6.3](#).

The mapping from one application schema to another application schema may be difficult and even impossible if the two schemas are too divergent. The mapping is facilitated if the application schema used within a system is designed considering also the requirements for the data interchange. The rules can also be used for building an application schema used within a system, although such application schemas are not within the scope of this document.

The creation of an application schema is a process. The content of an application schema has to be settled according to the view of reality in the universe of discourse. This is modelled in terms of types of features and their properties. [Clause 7](#) contains principles for consistently defining features.

The application schema defines the structure and content of data. It is expressed in a conceptual schema language (CSL). [Clause 7](#) also includes a model expressed in a UML profile that defines the concepts required to describe types of features. Feature type definitions may be documented in feature catalogues (ISO 19110). Such definitions may be used in an application schema. Other ISO geographic information standards define reusable modules of conceptual schemas that may be integrated in an application schema. [Clause 8](#) gives the general rules for integrating these predefined modules into a conceptual schema as main rules for defining them in UML.

NOTE ISO 19118:2011, ISO 19136:2020 and ISO/TS 19139-1:2019 define how a dataset defined by an application schema in UML can be encoded in XML.

6.3 Application schemas supporting data interchange

[Figure 1](#) shows the traditional data transfer model for data suppliers and data users. The structure and content of the data provided by the supplier and received by the user are described in application schemas. To be able to transfer data, it is necessary to fulfil three conditions.

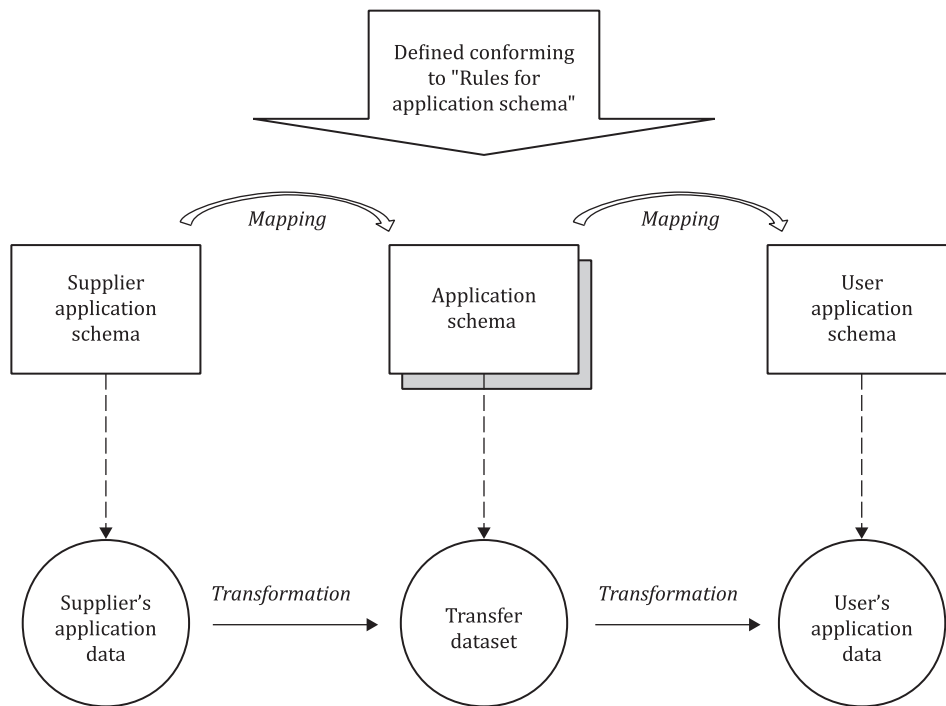


Figure 1 — Data interchange by transfer

First, the users and suppliers need to come to an agreement on creating an application schema for the data being interchanged in accordance with this document. In order to facilitate the interchange of data, the application schema is developed based on the application schemas from the users and the suppliers. Each supplier will create a mapping from its application schema to the common application schema for the interchanged data, and each user will create a mapping from the common application schema to the application schema of the user. This ensures that the meaning of the interchanged data is preserved and understood throughout the data interchanges between the involved parties. In other words, semantic interoperability is achieved. It also has the advantage that a new supplier or user only has to map its application schema onto the common application schema.

Second, the supplier needs to be capable of transforming the application data defined according to the supplier application schema into a transfer dataset defined according to the application schema for the exchanged data.

Third, the user needs to be able to transform the transfer dataset defined according to its application schema to the application data defined according to the user's application schema.

NOTE Conformance to the rules in this document does not guarantee that data conforming to any given application schema can be transformed in a meaningful way to conform to any other application schema. At best, it allows the user to determine which elements are common to the two schemas and which could be transformed from one schema to another, as well as those that cannot be transformed. Complete interoperability can only occur when user and supplier have identical application schemas.

7 General feature model

7.1 Principle for defining features

7.1.1 Features, coverages and properties

7.1.1.1 Features

A fundamental unit of geographic information is called a feature. ISO 19110 provides a standard framework for organizing and reporting the classification of features.

This document gives rules for creating application schemas, including the principles for the definition of features. The term “feature” is used in different contexts and defined according to the four-layer architecture. [Annex B](#) describes the use of the term feature in the four-layer architecture.

This document distinguishes three aspects of defining features: the definitions or description used to group them into types, the attributes associated with each type, and the relationships among the types.

EXAMPLE “Tower Bridge” is the abstraction of a certain real-world bridge in London. The term “bridge” is the abstraction of the collection of all real-world phenomena that is classified into the concept behind the term “bridge”. Later in the document, the terms “feature type” and “feature instance” are used to separate the concept of “feature” describing the whole collection from the concept of describing a certain instance.

[Figure 2](#) describes the most abstract level of defining and structuring geographic data. The classification of real-world phenomena as features depends on their significance to a particular universe of discourse.

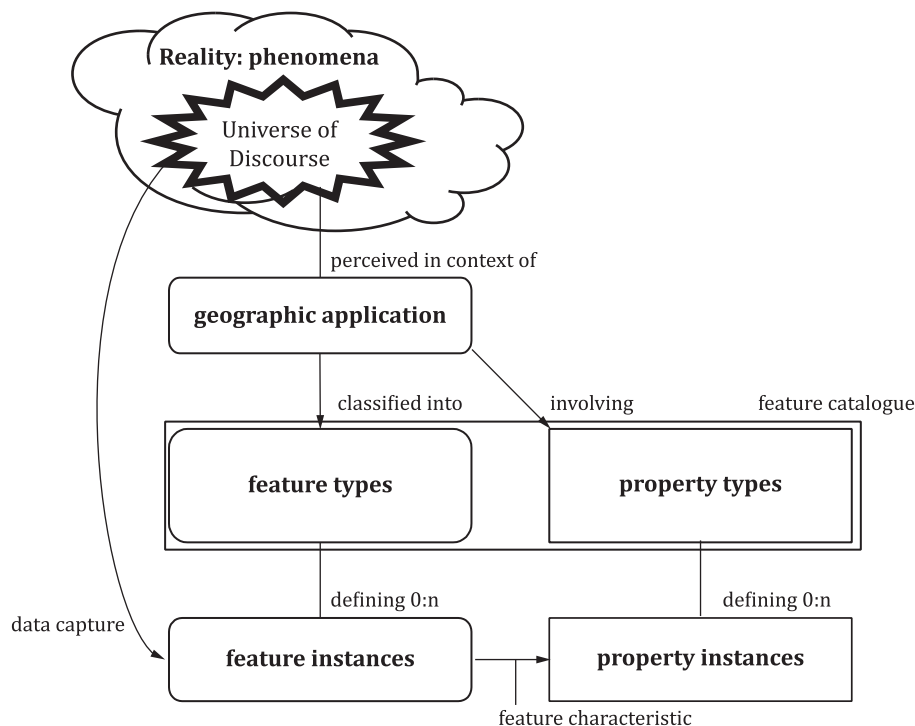


Figure 2 — The process from universe of discourse to data

7.1.1.2 Coverages

Many aspects of the real-world may be represented as features whose properties are single-valued and static. These conventional features provide a model of the world in terms of discrete objects located in it. However, in some applications it is more useful to use a model focusing on the variation of property values in space and time, formalized as coverages. Users of geographic information may utilize both viewpoints. While coverages are themselves strictly features as well, it is common to contrast coverages and non-

coverage features when discussing the functionality provided by each viewpoint. In the following discussion the name “feature” refers to non-coverage features.

The feature and coverage representations may be related in several ways:

- signal processing to find and characterize features: signals in coverages may provide evidence for the existence, location, and type of features, detected through modelling and interpretation;

EXAMPLE 1 Patterns of colour or other radiance bands within a remotely-sensed image can be used to infer the existence of specific objects or features on the ground.

EXAMPLE 2 Signals in a geophysical borehole log can be used to infer the presence of particular rock-units at underground locations.

- coverage-typed feature properties: feature properties whose value vary within the scope of a feature may be described as coverages whose domain extent is the geometry of the feature;

EXAMPLE 3 The variation of concentration of a particular ore-mineral within a mine can be described as a spatial function or coverage within the spatial limits of the mine.

- features sample a coverage: the values of a common property of a set of features provide a discrete sampling of a coverage, whose range type is the property, and whose domain is the aggregate geometry of the set of features.

EXAMPLE 4 The temperature at a set of weather stations can be compiled to show the spatial variation of temperature across the region where the stations are located.

A constraint in the latter two cases is that a property-type from a feature catalogue is the range-type of a coverage description in the same universe of discourse.

The case of features having property values that vary within the scope of the feature can be described using the general feature model ([7.4.8](#)).

While the coverage model is described in detail in ISO 19123-1, an application schema may include both feature- and coverage-types.

NOTE The feature and coverage viewpoints are related to (though not identical with) the so-called “vector” and “raster” approaches from traditional GIS implementations.

7.1.1.3 Properties and observations

Property values are associated with features and coverages. In the case of features, a property value is associated with a classified object. In the case of coverages, a property value is associated with a position in the domain.

Property values can be assigned in a number of ways, such as through assertion by a competent authority, by inheritance from a parent feature or type, by derivation from more primitive properties, or by observation. In the latter case the value assigned is often an estimate, with an error resulting from the observation procedure.

NOTE Following ISO 19156, “observation” denotes the act carried out by an observer to determine the value of an observable property of an object. The procedure may involve a sensor but can also be a computational or classification process.

The result of an observation is a property value that is assigned to its feature of interest. Where the feature property has a constant value, an associated observation result has a static value. Where the feature property varies with position, the result of an associated observation is a spatial function or coverage.

7.1.2 Essential properties of features

[Figure 3](#) describes the properties that are essential to describing a feature. Features can be described using three components namely, feature attributes, feature associations, and metadata properties. Feature attributes describe the spatial and non-spatial characteristics of a feature. Spatial attributes

cover geometric and topologic properties, thematic attributes carry its descriptive characteristics, and the temporal attributes represent the characteristics of the feature with respect to time. Feature associations describe the association relationships between features, which may also be spatial, thematic, or temporal in nature. Metadata properties carry the data about the features, such as quality information, which may be used for its discovery and identification. This structure, in turn, is the basis on how feature types and their associated attributes, and relationships contained in an application schema are defined and described in a feature catalogue.

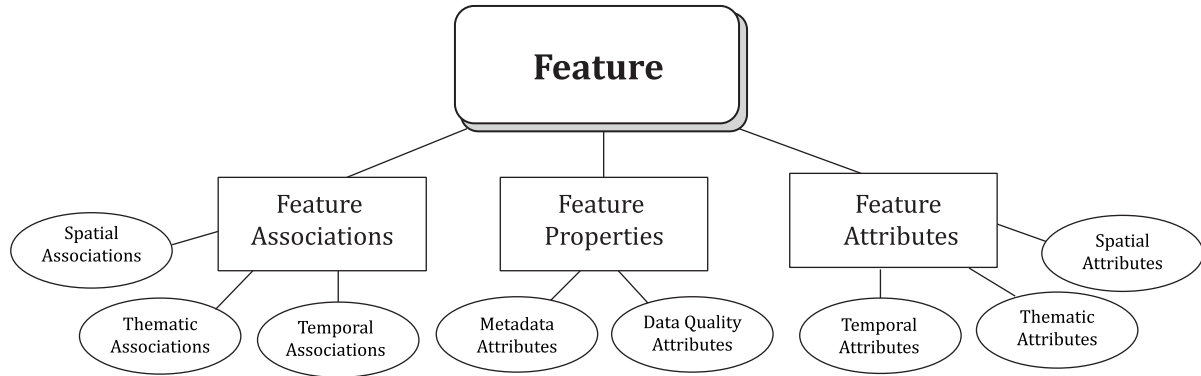


Figure 3 — Essential properties for defining features

7.2 The Concept of the General Feature Model

7.2.1 Introduction

Concepts used to define features and how these concepts are related are identified and described in 7.2.2 to 7.2.3. The description is expressed in a metamodel, also called the General Feature Model (GFM).

[Annex B](#) provides discussions regarding the purpose and design of the GFM.

[7.4](#) to [7.6](#) depict how different aspects of the properties of a feature are managed. [7.4](#) describes different aspects of attributes and [7.5](#) describes different aspects of relationships. [7.6](#) describes the concept of constraints.

7.2.2 The purpose of the GFM

The GFM is a model of the concepts required to classify a view of the real world. It is expressed in the Meta Object Facility (MOF) metamodeling language, which shares its metamodel with the Unified Modeling Language (UML). MOF 2.5.1 is a standard for defining and managing metadata in a unified way. The MOF offers an open, platform-independent framework for managing metadata and services that facilitate the development and interoperability of model-driven systems.

The things that need to be classified are called features. Feature types have properties that are feature attributes and feature association roles. The relations between feature types are feature associations and inheritance. All these concepts are expressed as metaclasses in the GFM. The GFM is a metamodel of feature types.

Feature types can be documented in feature catalogues. The GFM may serve as the conceptual model of the structure of a feature catalogue, but the feature catalogue has additional concepts for documenting feature types. There is a Feature Catalogue methodology that realizes the GFM concepts and adds more concepts, see ISO 19110. Some new concepts are, for example, a list of the feature attributes of each feature type, aliases of the feature type name, and codes for the feature type name. These additional concepts are not in conflict with the concepts from the GFM.

The GFM specifies the requirements for the classification of features but is not a CSL. We have to use an existing CSL to define the application schema. Within the ISO geographic information standards, UML is used to define abstract schemas. As the goal is to implement and integrate abstract schemas into application schemas, it is convenient to also express the application schema in UML. This document defines the main

rules for mapping GFM concepts into UML. While GFM can also be mapped to other CSLs, this is out of scope for this document. As UML provides a general mechanism for classifications of any kind, it has to be subsetting to be used as a basis for the classification of features. A useful subset of UML is specified in ISO 19103. ISO 19103 also specifies a UML profile, which is augmented by this document. The resulting UML profile is summarized in [8.10.2](#).

The GFM defines the structure for classifying features that we need to keep in mind when we make our application schemas in UML. However, the mapping from GFM to UML is a one-way mapping; it is not possible to map backwards. For example, the application schema has UML classes. Some of these classes are GFM feature types and some are the datatypes for feature attributes; it is not possible to keep these things apart. The GFM does not define feature-attribute values to the depth that is needed. That is not necessary to do as the GFM only specifies the structure and content of definition of features.

Hence, the GFM is a metamodel for definition of features that is also used to define the structure of feature catalogues. The chosen CSL is the metamodel for an application schema.

7.3 Conceptual Schema of the General Feature Model

7.3.1 The structure of the GFM

[Figure 4](#) describes the overall structure of the General Feature Model. The main part of the GFM contains the fundamental metaclasses that are essential for describing a feature. In the main part, some metaclasses are described in more detail by introducing specializations of them that describe more specific characteristics of the feature.

The main part of the GFM, represented in [Figure 5](#), is composed of the three major fundamental metaclasses, FeatureType, PropertyType, and ValueAssignment. The FeatureType metaclass represents the individual feature types, and the PropertyType metaclass represents the attributes for identifying and describing the feature. On the other hand, ValueAssignment metaclass is used to assign the PropertyType values to a FeatureType. PropertyType is the superclass for metaclasses AttributeType, and FeatureAssociationRole, each describing different characteristics of features.

The ValueAssignment part of the GFM is composed of the ValueAssignment metaclass and its corresponding instances. These instances describe specific methods to assign values to a property. The AttributeType part is composed of the child metaclasses of the AttributeType metaclass, which describe detailed spatial and non-spatial properties, based on other ISO geographic information standards. Finally, the FeatureAssociationType part contains FeatureAssociationRole metaclass, which is an aggregate of the FeatureAssociationType metaclass, having child metaclasses that describe different association types between feature types. [Sections 7.3.8, 7.4, and 7.5](#) discuss how these three parts are structured and details on the corresponding child metaclasses.

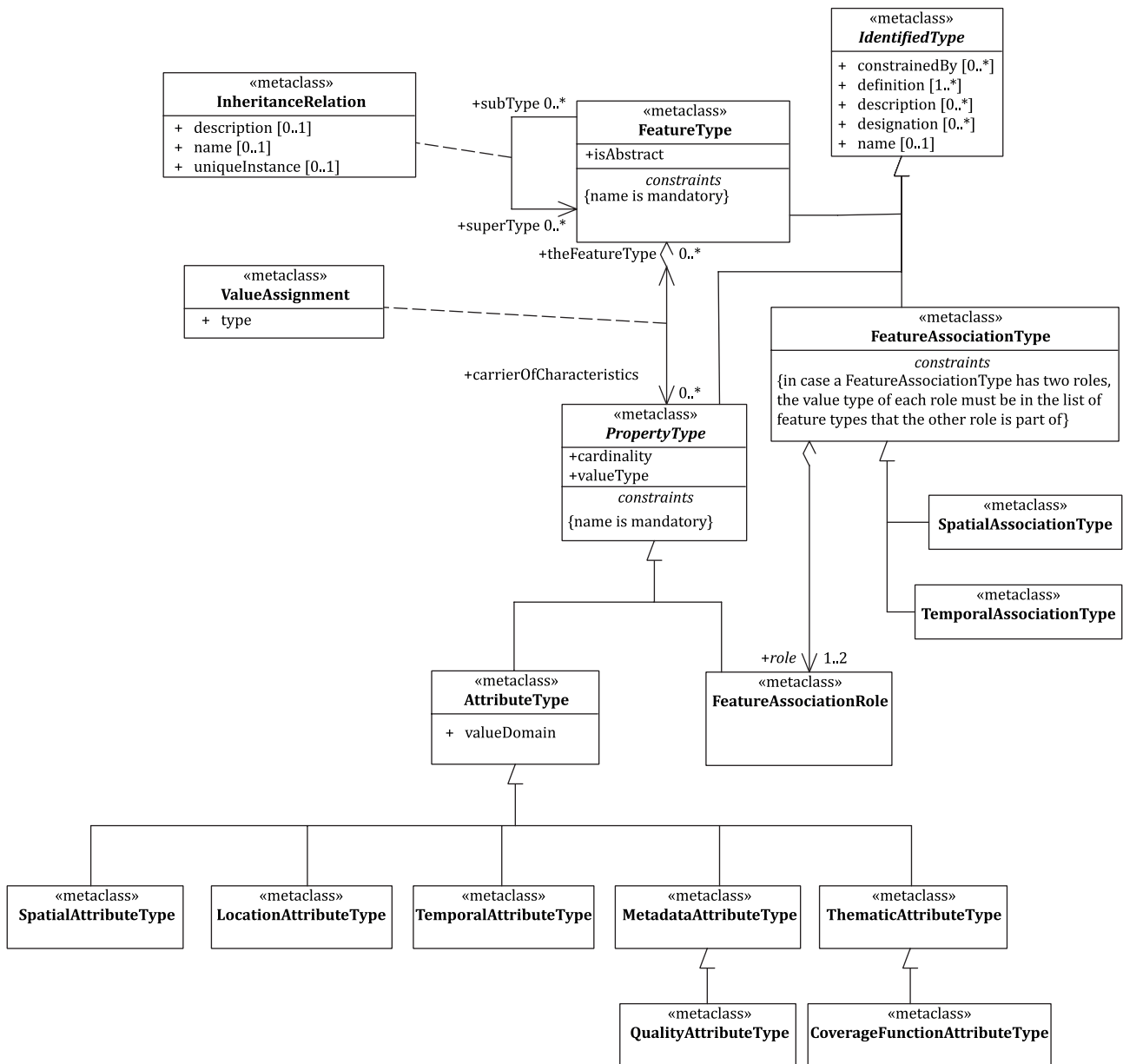


Figure 4 — Overview of the General Feature Model

7.3.2 The main part of the GFM

Figure 5 shows the concepts used to define types of features. Besides a name and a description, a feature type is defined by its properties such as:

- feature attributes;
- and feature association roles characterizing the feature type.

Additional concepts are:

- feature associations between the feature type and itself or other feature types;
- generalization and specialization relationships to other feature types;
- and constraints on the feature type.

NOTE Earlier versions of this document prefixed most class names with “GF_”. That convention is no longer used.

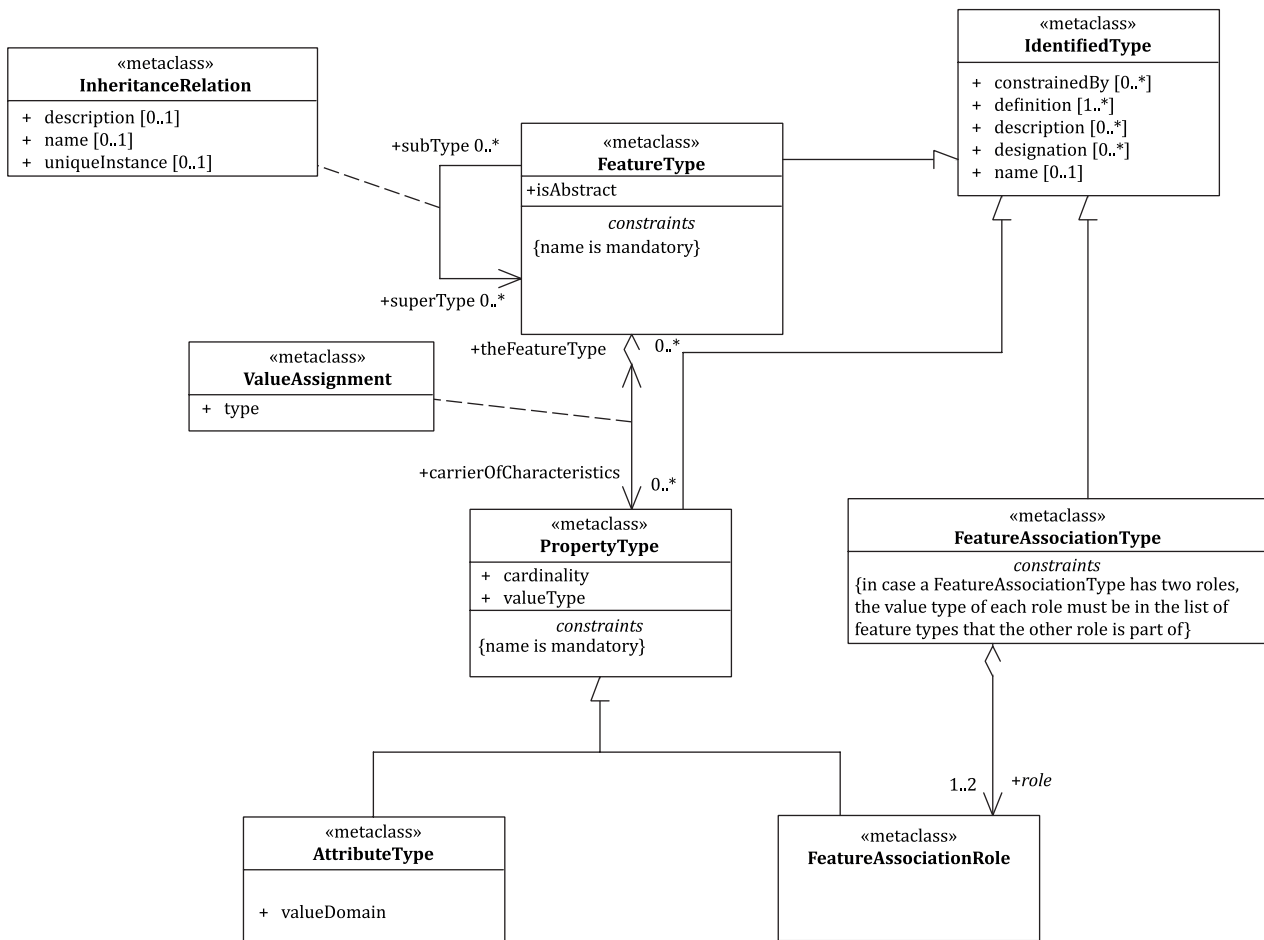


Figure 5 — Main part of the general feature model

7.3.3 Metaclass IdentifiedType

IdentifiedType is a metaclass from which the other key classes in the General Feature Model inherit some common identification and description attributes. *IdentifiedType* is the supertype of *FeatureType*, *PropertyType* and *FeatureAssociationType*. This metaclass is described in [Figure 5](#).

- name: a machine-readable name of the element, which is same in all languages that the schema is documented in.

The CSL may constrain the scope and namespace of model elements. In UML, the name of a class is scoped to the package, and the name of an attribute or role is scoped to a class, so when using UML as the CSL [\[Clause 8.10\]](#) the Application Schema provides the namespace for feature types, and the feature type provides the namespace for properties. In OWL, the namespace for both classes and properties are usually but not necessarily provided by the ontology. When using OWL as the CSL [\[ISO 19150-2\]](#) classes will typically be scoped to the ontology or application schema, but property types may be scoped to a class, to the application schema, or to an external ontology or catalogue.

- definition: representation of the element by an expression that describes it and differentiates it from other elements. One definition is mandatory. Additional definitions can be provided in multiple languages if required.
- designation: representation for the element by a sign which denotes it in a domain or subject.
- description: expression which serves to give information in addition to the definition.
- constrainedBy: description of a constraint made on the *IdentifiedType* or specified on properties within a type. Also see [7.6](#).

7.3.4 Metaclass *FeatureType*

A feature is an abstraction of real-world phenomena. *FeatureType* is a metaclass that is instantiated as classes that represent individual feature types. A certain feature type is the class for all instances of that feature type. The instances of a class that represents an individual feature type are feature instances.

FeatureType inherits attributes for identification and description from the abstract metaclass *IdentifiedType*. The name is mandatory on a feature type and is unique in an application schema.

FeatureType adds the following attributes and association ends:

- *isAbstract*: if true, the feature type acts as an abstract supertype.
- *carrierOfCharacteristics*: association role to specify any feature attribute type and any feature association role that carries characteristics of a feature type. While strictly optional, it is unlikely for a feature type to be defined having no properties. Repeatable.
- *superType*: association role to specify a supertype (generalized type) of the feature type (optional, repeatable).

7.3.5 Metaclass *PropertyType*

PropertyType is the abstract superclass for *AttributeType* and *FeatureAssociationRole*, which provide characteristics that may be associated with a feature type. *PropertyType* inherits attributes for identification and description from the abstract metaclass *IdentifiedType*. The inherited attribute *name* is mandatory for *PropertyType*.

Property types are specified independent of any specific feature type. A property instance and value occur in the context of a feature, though in some cases the feature is very generic and might not be made explicit in the dataset. For example, the feature type or feature identity for an observed property might not be available because the necessary data interpretation has not yet been completed. Properties will often be described independent of features. It is therefore best if property type names are unique within an application.

PropertyType adds the following attributes and association end:

- *cardinality*: number of instances of non-abstract subtypes of *PropertyType*. For the subtype *AttributeType* this is the number of instances of the attribute that may be associated with a single instance of a feature type. For the subtype *FeatureAssociationRole* this is the number of instances of the feature type that can act in this role relative to a single instance of the feature type at the other end of the association (mandatory);
- *valueType*: data type of the attribute value or the role value (mandatory);

EXAMPLE *Integer, CharacterString* or *Geometry*.

- *theFeatureType*: association role to specify the feature types that carry this property.

7.3.6 Metaclass *AttributeType*

AttributeType is the metaclass for attribute definitions of a feature type (see also [7.4](#)). *AttributeType* inherits attributes and associations from the abstract metaclass *PropertyType*.

AttributeType adds the following attribute value.

- *valueDomain*: description of a set of values. In some cases, the domain of values may be a set of concepts, organized in a vocabulary or lexicon.

EXAMPLE 1 Positive, from 3 to 7; *Geometry* and all its subtypes as defined in ISO 19107; enumerated set.

EXAMPLE 2 An attribute that carries the position of a feature can have another attribute that holds the positional accuracy (data value of *QualityAttributeType*) of this position.

A taxonomy of attribute types is described in [7.4](#).

7.3.7 Metaclass FeatureAssociationRole

FeatureAssociationRole is the metaclass for the classes of roles that are part of a *FeatureAssociationType* (7.5.3). *FeatureAssociationRole* indicates the role played by the feature type through the association. The instance of *FeatureAssociationRole* that gives the role for one feature type can also be seen as part of this feature type. *FeatureAssociationRole* inherits attributes and associations from the abstract metaclass *PropertyType*.

7.3.8 Metaclass ValueAssignment

PropertyType is the metaclass for any class of property of a feature type that describes characteristics of the feature, the behaviour of a feature, or the association roles that the feature is in. *PropertyType* is the supertype of *AttributeType* and *FeatureAssociationRole*.

Values can be assigned to properties through a variety of methods. In many applications and for many properties the method is either not known or is not of sufficient interest for it to be explicitly modelled. Nevertheless, in principle a value assignment process is associated with every property value. *ValueAssignment* is the association metaclass for any process by which values are assigned to a property, and when required, instances of this metaclass support auditing the evidence for assignment of a property value.

The feature type *Observation* (ISO 19156), and the classes *LI_ProcessStep* (ISO 19115-1) and *MI_Event* (ISO 19115-2) may be interpreted as instances of *ValueAssignment* (Figure 6). The description of a specific observation can provide an evaluation of the likely error in a property value.

NOTE The *Observation* class in ISO 19156 carries an explicit constraint that the feature-property combination is consistent.

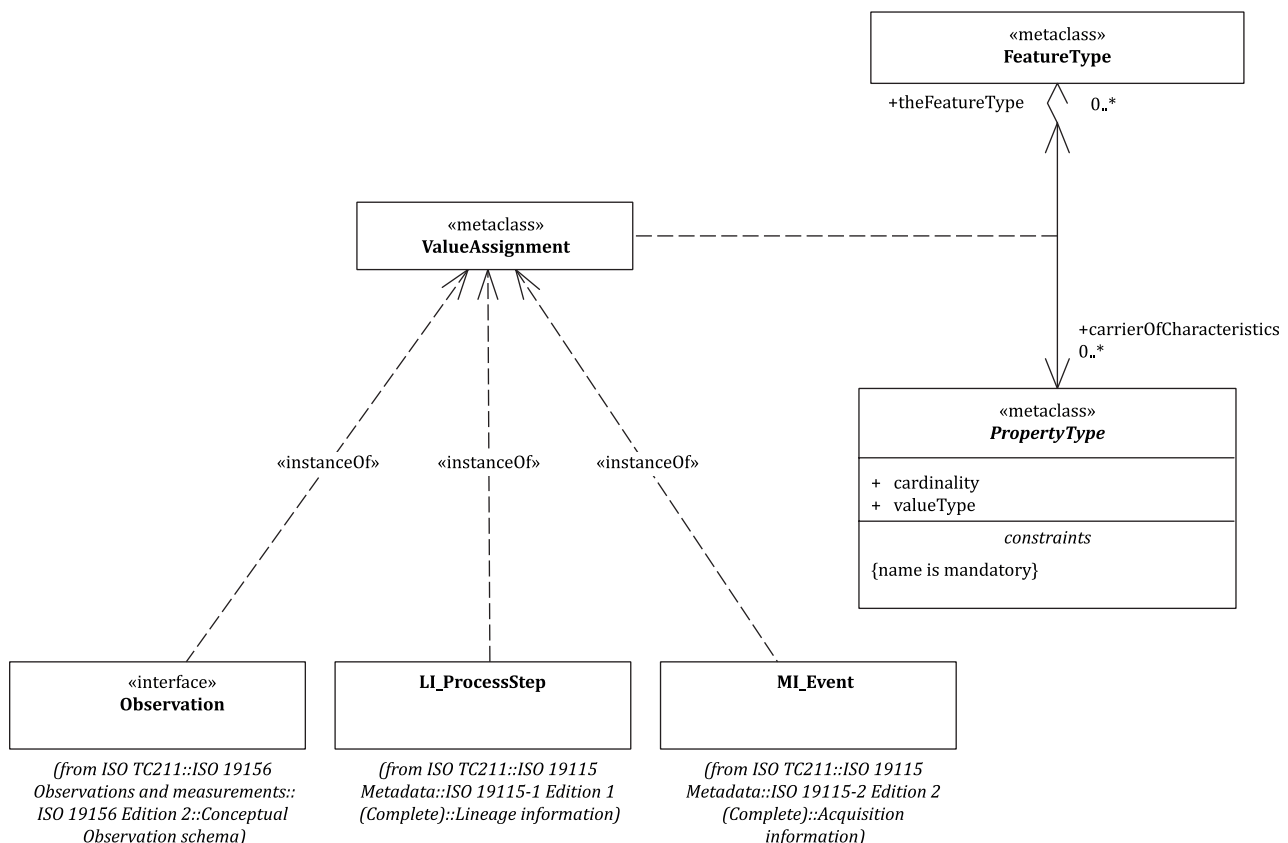


Figure 6 — Example instances of *ValueAssignment* metaclass

Other examples of value assignment classes are:

- assertion, in which a property value is assigned by a competent authority;