

DIN EN ISO 19109

ICS 35.240.70

Einsprüche bis 2014-01-15
Vorgesehen als Ersatz für
DIN EN ISO 19109:2006-09

Entwurf

**Geoinformation –
Regeln zur Erstellung von Anwendungsschemata (ISO/DIS 19109:2013);
Englische Fassung prEN ISO 19109:2013**

Geographic information –
Rules for application schema (ISO/DIS 19109:2013);
English version prEN ISO 19109:2013

Information géographique –
Règles de schéma d'application (ISO/DIS 19109:2013);
Version anglaise prEN ISO 19109:2013

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2013-11-15 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an nabau@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den Normenausschuss Bauwesen (NABau) im DIN, 10772 Berlin (Hausanschrift: Burggrafenstr. 6, 10787 Berlin).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 113 Seiten

Normenausschuss Bauwesen (NABau) im DIN

Nationales Vorwort

Dieses Dokument (prEN ISO 19109:2013) wurde vom Technischen Komitee ISO/TC 211 „Geographic information/Geomatics“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 287 „Geoinformation“, dessen Sekretariat von BSI (Vereinigtes Königreich) gehalten wird, erarbeitet.

Der für die deutsche Mitarbeit zuständige Arbeitsausschuss im DIN Deutsches Institut für Normung e. V. ist der als Spiegelausschuss zum CEN/TC 287 und ISO/TC 211 eingesetzte Arbeitsausschuss NA 005-03-03 AA „Kartographie und Geoinformation“ des Normenausschusses Bauwesen (NABau).

Dieses Dokument ist derzeit zur Umfrage vorgelegt.

Das Präsidium des DIN hat mit seinem Beschluss 1/2004 festgelegt, dass von dem in den Regeln der europäischen Normungsarbeit von CEN/CENELEC verankerten Grundsatz, wonach Europäische Normen in den drei offiziellen Sprachen Deutsch, Englisch und Französisch veröffentlicht werden, in begründeten Ausnahmefällen abgewichen und auf die deutsche Sprachfassung verzichtet werden kann.

Die Genehmigung dafür hat die DIN-Geschäftsleitung entsprechend den festgelegten Kriterien für die vorliegende Norm auf Antrag des Normenausschusses Bauwesen (NABau), Fachbereich 03 „Vermessungswesen; Geoinformation“ erteilt.

Änderungen

Gegenüber DIN EN ISO 19109:2006-09 wurden folgende Änderungen vorgenommen:

- a) Klarere Struktur der Konformitätsklassen im Einklang mit den entsprechenden Empfehlungen von ISO/TC 211;
- b) Berücksichtigung von Metadaten zur Bestimmung von Eigenschaften eines Features nach DIN EN ISO 19156, *Geoinformation — Erdbeobachtung und Erdmessung*;
- c) Anpassung an die überarbeitete DIN EN ISO 19103 und Definition eines UML-Profiles für DIN EN ISO 19109;
- d) Anpassung des General Feature Model an die Nutzung in der Praxis;
- e) generelle Überarbeitung für eine klarere Darstellung der Anforderungen und Empfehlungen.

Nationaler Anhang NA (informativ)

Begriffe

Die Benennungen der folgenden Begriffe ist identisch mit den Benennungen in der Englischen Fassung.

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

4.1

Anwendung

(en: application)

Be- und Verarbeitung von Daten zur Unterstützung von Nutzeranforderungen

4.2

Anwendungsschema

(en: application schema)

konzeptuelles Schema für Daten, das für eine oder mehrere Anwendungen erforderlich ist

4.3

komplexes Feature

(en: complex feature)

Feature, das sich aus anderen Objekten zusammensetzt

4.4

konzeptuelles Modell

(en: conceptual model)

Modell, das einen abstrahierten Bereich der realen Welt abbildet

4.5

konzeptuelles Schema

(en: conceptual schema)

formale Beschreibung eines konzeptuellen Modells

4.6

Coverage

(en: coverage)

Feature, das in Form einer Funktion für jeden räumlichen, zeitlichen oder raum-zeitlichen Punkt innerhalb seines Definitionsbereichs einen Wertevektor liefert

4.7

Datensatz

(en: dataset)

identifizierbare Sammlung von Daten

4.8

Bereich

(en: domain)

wohldefiniertes (Fach-) Gebiet oder bestimmter Wertebereich

4.9

Feature

(en: feature)

Abstraktion eines Phänomens der realen Welt

4.10**Feature-Assoziation**

(en: feature association)

Beziehung, die Instanzen einer Feature-Klasse mit Instanzen derselben oder einer anderen Feature-Klasse verbindet

4.11**Feature-Attribut**

(en: feature attribute)

Merkmal eines Features

4.12**Feature-Operation**

(en: feature operation)

Operation, die jede Instanz einer Feature-Klasse ausführen kann

4.13**geographische Daten**

(en: geographic data)

Daten mit einem impliziten oder expliziten Bezug zu einem Ort auf der Erde

4.14**Metadaten**

(en: meta data)

Information über eine Ressource

4.15**Modell**

(en: model)

Abstraktion einiger Aspekte der Realität

4.16**Beobachtung**

(en: observation)

Messvorgang oder anderer Vorgang zur Ermittlung des Werts einer Eigenschaft

4.17**Eigenschaft**

(en: property)

benannter Aspekt oder benanntes Attribut eines Objekts

4.18**Qualität**

(en: quality)

Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt

4.19**Gegenstandsbereich**

Ausschnitt der realen Welt

(en: universe of discourse)

Sicht der realen oder hypothetischen Welt, die die Gesamtheit dessen umfasst, was von Interesse ist

4.20**Wert**

(en: value)

Element eines Typs und Bereichs

Contents

Page

Foreword	vii
Introduction.....	viii
1 Scope	1
2 Conformance	1
2.1 General	1
2.2 Meta-model.....	2
2.3 UML application schema	2
2.4 Profiling standard schema	2
2.5 Metadata	2
2.6 Quality.....	2
2.7 Temporal.....	3
2.8 Spatial	3
2.9 Coverages	3
2.10 Observations.....	3
2.11 Spatial referencing by identifiers.....	4
2.12 Code-list	4
2.13 Multi-lingual support	4
3 Normative references	4
4 Terms and definitions	5
5 Presentation and abbreviations	7
5.1 Presentation	7
5.1.1 General	7
5.1.2 Conformance class	7
5.1.3 Requirements class.....	8
5.1.4 Rules	8
5.1.5 Identifiers.....	8
5.1.6 Conceptual schemas.....	9
5.1.7 Descriptions of concepts.....	9
5.2 Abbreviations.....	9
5.3 Package abbreviations.....	9
6 Context	10
6.1 Purpose of an application schema	10
6.2 Rules for application schema	10
6.3 Application schema supporting data interchange.....	11
6.3.1 Introduction.....	11
6.3.2 Data interchange by transfer.....	11
6.3.3 Data interchange by transactions.....	12
7 Principles for defining features	13
7.1 General	13
7.2 Features, Coverages and Properties	14
7.2.1 Features.....	14
7.2.2 Coverages	15
7.2.3 Properties and observations	16
7.3 Features and the application schema	16
7.4 The General Feature Model	17
7.4.1 Introduction.....	17
7.4.2 The purpose of the GFM	18
7.4.3 The main structure of the GFM	18

7.4.4	IdentifiedType.....	19
7.4.5	FeatureType.....	20
7.4.6	PropertyType.....	21
7.4.7	AttributeType.....	21
7.4.8	Operation	22
7.4.9	FeatureAssociationRole.....	22
7.4.10	ValueAssignment.....	23
7.4.11	FeatureAssociationType	25
7.4.12	InheritanceRelation.....	25
7.5	Attributes of feature types	26
7.5.1	Introduction	26
7.5.2	SpatialAttributeType.....	27
7.5.3	TemporalAttributeType	27
7.5.4	QualityAttributeType	27
7.5.5	LocationAttributeType	27
7.5.6	MetadataAttributeType.....	28
7.5.7	ThematicAttributeType.....	28
7.5.8	CoverageFunctionAttributeType.....	28
7.6	Relationships between feature types	28
7.6.1	Introduction	28
7.6.2	InheritanceRelation.....	28
7.6.3	FeatureAssociationType	29
7.7	Constraints	30
8	Rules for application schema in UML.....	30
8.1	The application modelling process.....	30
8.2	The application schema	31
8.2.1	General.....	31
8.2.2	Conceptual schema language for application schemas	32
8.2.3	Packaging and identification of an application schema.....	33
8.2.4	Documentation of an application schema	33
8.2.5	Integration of application schemas and standard schemas	34
8.2.6	Modelling structures in UML	36
8.3	Domain profiles of standard schemas in UML	41
8.3.1	Introduction	41
8.3.2	Adding information to a standard schema	41
8.3.3	Tailored use of standard schemas.....	42
8.4	Rules for use of metadata schema	43
8.4.1	Introduction	43
8.4.2	Metadata for features, feature attributes, and feature associations	43
8.5	Rules for use of quality schema.....	44
8.5.1	Introduction	44
8.5.2	Data quality rules	45
8.6	Temporal rules	48
8.6.1	Rules for modelling applications with temporal properties	48
8.6.2	Use of the temporal conceptual schema.....	49
8.6.3	Temporal attributes	50
8.6.4	Temporal associations between features	52
8.7	Spatial rules.....	55
8.7.1	Rules for modelling applications with spatial properties.....	55
8.7.2	Use of standard spatial schema.....	56
8.7.3	Spatial attributes	57
8.7.4	Use of geometric aggregates and spatial complexes to represent the values of spatial attributes of features	58
8.7.5	Spatial associations between features.....	63
8.7.6	Features sharing geometry.....	64
8.7.7	Point features, line features and area features.....	65
8.7.8	Defining interpolation methods.....	65
8.7.9	Independent spatial complexes	66
8.8	Rules for use of coverage functions.....	68

8.9	Rules for the use of observations	71
8.10	Spatial referencing using geographic identifiers	75
8.11	Code-lists, vocabularies, lexicons	77
8.12	Linguistic adaptation	78
Annex A	(normative) Abstract test suite	80
A.1	Introduction	80
A.2	Meta-model	80
A.2.1	Conceptual schema language	80
A.2.2	Schema integration	80
A.2.3	Features	80
A.2.4	Value assignment	81
A.2.5	Feature associations	81
A.2.6	Feature inheritance	81
A.2.7	Constraints	82
A.3	Creating application schemas in UML	82
A.3.1	UML profile	82
A.3.2	Packaging	82
A.3.3	Documentation	82
A.3.4	Integrating an application schema	83
A.3.5	Modelling structure	83
A.3.6	Modelling features	83
A.3.7	Property names	84
A.3.8	Value assignment	84
A.3.9	Attribute of attribute	84
A.3.10	Feature operation	84
A.3.11	Feature inheritance	85
A.3.12	Constraints	85
A.4	Profiling standard schemas	85
A.4.1	Extension	85
A.4.2	Restriction	86
A.5	Use of metadata schema	86
A.6	Use of quality schema	86
A.6.1	Use of the quality schema in an application schema	86
A.6.2	Additional quality metadata in an application schema	87
A.6.3	Instance quality metadata in an application schema	87
A.7	Temporal elements	87
A.7.1	Use of the temporal conceptual schema in an application schema	87
A.7.2	Temporal attributes	87
A.7.3	Temporal associations	88
A.7.4	Feature succession	88
A.8	Spatial elements	88
A.8.1	Use of the spatial conceptual schema in an application schema	88
A.8.2	Spatial attributes	89
A.8.3	Spatial complexes	89
A.8.4	Spatial associations	89
A.8.5	Shared geometry	89
A.8.6	Single geometry	90
A.8.7	Interpolation method	90
A.8.8	Independent spatial complexes	90
A.9	Coverages	91
A.10	Observations	91
A.11	Spatial referencing using geographic identifiers	91
A.12	Code-lists and vocabularies	92
A.12.1	Management of code-lists	92
A.12.2	Formalization of code-lists	92
A.13	Multi-lingual support	92
A.13.1	Application schema	92
A.13.2	Feature and property types	92
Annex B	(informative) The modelling approach and the General Feature Model	94

B.1	The four-layer architecture	94
B.1.1	Introduction	94
B.1.2	The Meta Meta level and the Meta level	95
B.1.3	The Meta level and the Application level	95
B.1.4	The Application level and the Data level	95
B.1.5	The General Feature Model and the conceptual schema language of the application schema.....	95
B.2	The term “feature”	96
Annex C	(informative) Application schema examples.....	97
C.1	Utility network	97
C.1.1	Introduction	97
C.1.2	The application schema	98
C.1.3	An example of the document on Utility Network.....	99
C.2	Administrative units	101
C.2.1	Introduction	101
C.2.2	The application schema in UML.....	102
Bibliography		103

Tables

Table 1	— Meta-model conformance class	2
Table 2	— UML application schema conformance class.....	2
Table 3	— Profiling standard schema conformance class.....	2
Table 4	— Metadata conformance class.....	2
Table 5	— Quality conformance class	2
Table 6	— Temporal conformance class	3
Table 7	— Spatial conformance class	3
Table 8	— Coverages conformance class.....	3
Table 9	— Observations conformance class	3
Table 10	— Spatial referencing by identifiers conformance class	4
Table 11	— Code-list conformance class.....	4
Table 12	— Multi-lingual support conformance class	4
Table 13	— Conformance class template.....	8
Table 14	— Requirements class template.....	8
Table 15	— Requirements class covering general requirements and the meta-model for application schemas	14
Table 16	— Requirements class for modelling an application schema in UML.....	31
Table 17	— Summary of UML Profile used as the CSL for application schemas	33
Table 18	— Example of real-world concepts in terms of the General Feature Model	40
Table 19	— Requirements class for defining a profile of a standard schema.....	41
Table 20	— Requirements class for an application schema including metadata	43
Table 21	— Requirements class for an application schema including metadata	45
Table 22	— Concrete subtypes of DQ_Element	46
Table 23	— Requirements class for applications with temporal properties.....	49
Table 24	— List of valid temporal objects for temporal attributes in an application schema	50
Table 25	— Requirements class for applications with spatial properties	56
Table 26	— List of valid spatial objects for spatial attributes in an application schema	57
Table 27	— Requirements class for applications with coverage-valued properties	69
Table 28	— List of valid coverage types in an application schema	69
Table 29	— Requirements class for applications involving observations	73
Table 30	— Requirements class for applications using geographic identifiers	76
Table 31	— Requirements class for applications schemas using code-lists.....	78
Table 32	— Requirements class for application schemas in multi-lingual environments.....	78
Table B.1	— Feature used in different levels	96

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

ISO 19109 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

This second edition cancels and replaces the first edition (ISO 19109:2005), which has been technically revised.

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, the data structures for data access and exchange must be fully documented. The interfaces between systems, therefore, need to be defined with respect to data and operations, using the methods standardized in this International Standard. For the construction of internal software and data storage within proprietary systems, any method may be used that enables the standardized interfaces to be supported.

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.

Geographic information — Rules for application schema

1 Scope

This International Standard defines rules for creating and documenting application schemas, including principles for the definition of features.

The scope of this International Standard includes the following:

- conceptual modelling of features and their properties from a universe of discourse;
- definition of application schemas;
- use of the conceptual schema language for application schemas;
- 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.

The following are outside the scope:

- choice of one particular conceptual schema language for application schemas;
- definition of any particular application schema;
- 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 Conformance

2.1 General

This International Standard defines 12 conformance classes shown in Tables 1 to 12, matching the 12 requirements classes described in Clauses 7 and 8. Any application schema claiming conformance to any requirements class in this International Standard 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.

2.2 Meta-model

Table 1 — Meta-model conformance class

Conformance class	/conf/general
Requirements	/req/general (Clause 7, Table 15)
Tests	All tests in A.2

2.3 UML application schema

Table 2 — UML application schema conformance class

Conformance class	/conf/uml
Dependency	/conf/general (2.2)
Requirements	/req/uml (8.2, Table 16)
Tests	All tests in A.3

2.4 Profiling standard schema

Table 3 — Profiling standard schema conformance class

Conformance class	/conf/profile
Dependency	/conf/uml (2.3)
Requirements	/req/profile (8.3, Table 19)
Tests	All tests in A.4

2.5 Metadata

Table 4 — Metadata conformance class

Conformance class	/conf/metadata
Dependency	/conf/uml (2.3)
Requirements	/req/metadata (8.4, Table 20)
Tests	All tests in A.5

2.6 Quality

Table 5 — Quality conformance class

Conformance class	/conf/quality
Dependency	/conf/uml (2.3)
Requirements	/req/quality (8.5, Table 21)
Tests	All tests in A.6

2.7 Temporal

Table 6 — Temporal conformance class

Conformance class	/conf/temporal
Dependency	/conf/uml (2.3)
Requirements	/req/temporal (8.6, Table 23)
Tests	All tests in A.7

2.8 Spatial

Table 7 — Spatial conformance class

Conformance class	/conf/spatial
Dependency	/conf/uml (2.3)
Requirements	/req/spatial (8.7, Table 25)
Tests	All tests in A.8

2.9 Coverages

Table 8 — Coverages conformance class

Conformance class	/conf/coverage
Dependency	/conf/uml (2.3)
Requirements	/req/coverage (8.8, Table 27)
Tests	All tests in A.9

2.10 Observations

Table 9 — Observations conformance class

Conformance class	/conf/observation
Dependency	/conf/uml (2.3)
Requirements	/req/observation (8.9, Table 29)
Tests	All tests in A.10

2.11 Spatial referencing by identifiers

Table 10 — Spatial referencing by identifiers conformance class

Conformance class	/conf/identifier
Dependency	/conf/uml (2.3)
Requirements	/req/identifier (8.10, Table 30)
Tests	All tests in A.11

2.12 Code-list

Table 11 — Code-list conformance class

Conformance class	/conf/codeList
Dependency	/conf/uml (2.3)
Requirements	/req/codeList (8.11, Table 31)
Tests	All tests in A.12

2.13 Multi-lingual support

Table 12 — Multi-lingual support conformance class

Conformance class	/conf/multi-lingual
Dependency	/conf/uml (2.3)
Requirements	/req/multi-lingual (8.12, Table 32)
Tests	All tests in A.13

3 Normative references

The following referenced documents are indispensable for the application 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.

IETF RFC 5646 (2009), Tags for identifying languages

ISO 19103:—¹⁾, *Geographic information — Conceptual schema language*

ISO 19107:2003, *Geographic information — Spatial schema*

ISO 19108:2002, *Geographic information — Temporal schema*

ISO 19112:2003, *Geographic information — Spatial referencing by geographic identifiers*

1) To be published.

ISO 19115-1:—²⁾, *Geographic information — Metadata — Part 1: Fundamentals*

ISO 19115-2:2009, *Geographic information — Metadata — Part 2: Extensions for imagery and gridded data*

ISO 19123:2005, *Geographic information — Schema for coverage geometry and functions*

ISO 19156:2011, *Geographic information — Observations and measurements*

ISO 19157:—³⁾, *Geographic information — Data Quality*

ISO 19505-2:2012, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure (Version 2.4.1)*

4 Terms and definitions

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

4.1

application

manipulation and processing of data in support of user requirements

[SOURCE: ISO 19101-1:—⁴⁾, 4.1.1]

4.2

application schema

conceptual schema (4.5) for data required by one or more *applications* (4.1)

[SOURCE: ISO 19101-1:—⁴⁾, 4.1.2]

4.3

complex feature

feature (4.9) composed of other features

4.4

conceptual model

model (4.15) that defines concepts of a *universe of discourse* (4.19)

[SOURCE: ISO 19101-1:—⁴⁾, 4.1.5]

4.5

conceptual schema

formal description of a *conceptual model* (4.4)

[SOURCE: ISO 19101-1:—⁴⁾, 4.1.6]

4.6

coverage

feature (4.9) that acts as a function to return *values* (4.20) from its range for any direct position within its spatial, temporal or spatiotemporal *domain* (4.8)

[SOURCE: ISO 19123:2005, 4.1.7]

2) To be published

3) To be published.

4) To be published.

4.7

dataset

identifiable collection of data

[SOURCE: ISO 19115-1:—⁴) 4.3]

4.8

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.

[SOURCE: ISO/TS 19103:2005, 4.1.6]

4.9

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 should be used when only one is meant.

[SOURCE: ISO 19101-1:—⁴), 4.1.11]

4.10

feature association

relationship that links instances of one *feature* (4.9) type with instances of the same or a different feature type

[SOURCE: ISO 19110:2005, 4.2]

4.11

feature attribute

characteristic of a *feature* (4.9)

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

Note 2 to entry: A feature attribute type has a name, a data type and a *domain* (4.8) associated with it. A feature attribute instance has an attribute *value* (4.20) taken from the domain of the feature attribute type.

[SOURCE: ISO 19101-1:—⁴), 4.1.12, modified – Notes and examples in 19101-1 have been omitted and other notes have been added.]

4.12

feature operation

operation that every instance of a *feature* (4.9) type may perform

EXAMPLE 1 A feature operation upon the feature type “dam” is to raise the dam. The results of this operation are to raise the height of the “dam” and the level of water in a “reservoir”.

EXAMPLE 2 A feature operation by the feature type “dam” might be to block vessels from navigating along a watercourse.

[SOURCE: ISO 19110:2005, 4.5, modified - The note given in ISO 19110:2005 for this entry has been omitted. A second example has been added.]

4.13

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.

4.14

metadata

information about a resource

[SOURCE: ISO 19115-1:—⁴), 4.10]

4.15

model

abstraction of some aspects of reality

4.16

observation

act of measuring or otherwise determining the *value* (4.20) of a *property* (4.17)

[SOURCE: ISO 19156:2011, 4.11]

4.17

property

facet or attribute of an object referenced by a name

[SOURCE: ISO 19143:2010, 4.21]

4.18

quality

degree to which a set of inherent characteristics fulfils requirements

[SOURCE: ISO 9000:2005, 3.1.1]

4.19

universe of discourse

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

[SOURCE: ISO 19101-1:—⁴), 4.1.38]

4.20

value

element of a type and *domain* (4.8)

[SOURCE: ISO/IEC 19505-2:2012, N.M]

5 Presentation and abbreviations

5.1 Presentation

5.1.1 General

This International Standard 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 International Standard provides guidance through examples.

5.1.2 Conformance class

Conformance to this International Standard is possible at a number of levels, specified by conformance classes (Clause 2). Each conformance class is summarized using the template shown as Table 13.

Table 13 — Conformance class template

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

All tests in a class must be passed, so dependencies are on other conformance classes⁵⁾. Each conformance class tests conformance to a set of requirements packaged in a requirements class (Clauses 7 and 8).

5.1.3 Requirements class

Each normative statement (requirement or recommendation) in this International Standard is a member of a requirements class. In this International Standard each requirements class is described in a discrete clause or sub-clause, and summarized using the template shown as Table 14.

Table 14 — Requirements class template

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

All requirements in a class must be satisfied, so the requirements class is the unit of re-use and dependency. Hence, the value of a Dependency requirement is another requirements class.

5.1.4 Rules

All rules are normative, and each rule is presented using the following template:

/req/{classM}/{reqN}	[Normative statement]
----------------------	-----------------------

where /req/{classM}/{reqN} identifies the requirement or recommendation. The use of this layout convention allows the normative provisions of this International Standard 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 a complete URI which allows the requirements class, requirement or recommendation to be identified in an external context. For example, following the URI scheme for ISO standards [IETF RFC 5141] a URI denoting this standard is

⁵⁾ See ISO/TC 211 Resolution 570.

<http://standards.iso.org/iso/19109/2>

The URI for each requirements class has the form

[http://standards.iso.org/iso/19109/2/req/\[classM\]](http://standards.iso.org/iso/19109/2/req/[classM])

and the URI for each requirement or recommendation has the form

[http://standards.iso.org/iso/19109/2/req/\[classM\]/\[reqN\]](http://standards.iso.org/iso/19109/2/req/[classM]/[reqN])

5.1.6 Conceptual schemas

Conceptual schemas in the normative part of this International Standard are presented in the Unified Modeling Language (UML) in conformance with ISO 19103:201X. UML diagrams are presented in compliance with ISO/IEC 19505-2.

5.1.7 Descriptions of concepts

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

5.2 Abbreviations

CSL	Conceptual schema language
GFM	General feature model
OCL	Object constraint language
OWL	Web Ontology Language
UML	Unified Modeling Language

5.3 Package abbreviations

Concepts from schemas defined in some other International Standards are designated with names that start with bi-alpha codes as follow:

GM	ISO 19107:2003 Spatial Schema
TM	ISO 19108:2002 Temporal Schema, Temporal Objects
TP	ISO 19108:2002 Temporal Schema, Temporal Topology
MD	ISO 19115 Metadata
CV	ISO 19123:2005 Schema for Coverage Geometry and Functions
OM	ISO 19156:2011 Observations and Measurements
DQ	ISO 19157:201X Data Quality

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
- operations for manipulating and processing 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;
- to 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 Rules for application schema

This International Standard 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 International Standard 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, and
- determine the necessary transformations between the two datasets.

The rules in this International Standard 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 International Standard.

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 19110:2005). 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 main rules for integrating these predefined modules into a conceptual schema in UML.

NOTE ISO 19118:2005, ISO 19136:2007 and ISO/TS 19139:2007 define how a dataset defined by an application schema in UML is encoded.

6.3 Application schema supporting data interchange

6.3.1 Introduction

Data interchange between information systems may take place in two ways:

- In the traditional data transfer model, the data supplier creates a dataset that is transferred to the user. The structure and the content of data are described in the application schema for the dataset. The dataset is sent in a transfer format;
- In the interoperability model, the user application communicates with the supplier application through a common communication protocol. In this scenario, the user invokes services that result in data being passed from the service provider to the user application. The application schema describes not only the structure and content of the exchanged data, but also the structure of the interfaces involved in the transaction.

There is a fundamental distinction between a data transfer and a data transaction. In data transfer, a data set is predefined in an application schema. The spatial extent and the rules for inclusion of feature instances are also predefined. The user requests and receives a copy of the dataset (or may receive it automatically through a long-term agreement for distribution of datasets). In a data transaction, a requester first specifies selection criteria, such as spatial extent and feature instance inclusion rules for the data from the producer's data store. Data meeting the selection criteria are then retrieved from the data store and provided to the user.

NOTE Conformance to the rules in this International Standard 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.

6.3.2 Data interchange by transfer

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, three conditions must be fulfilled.

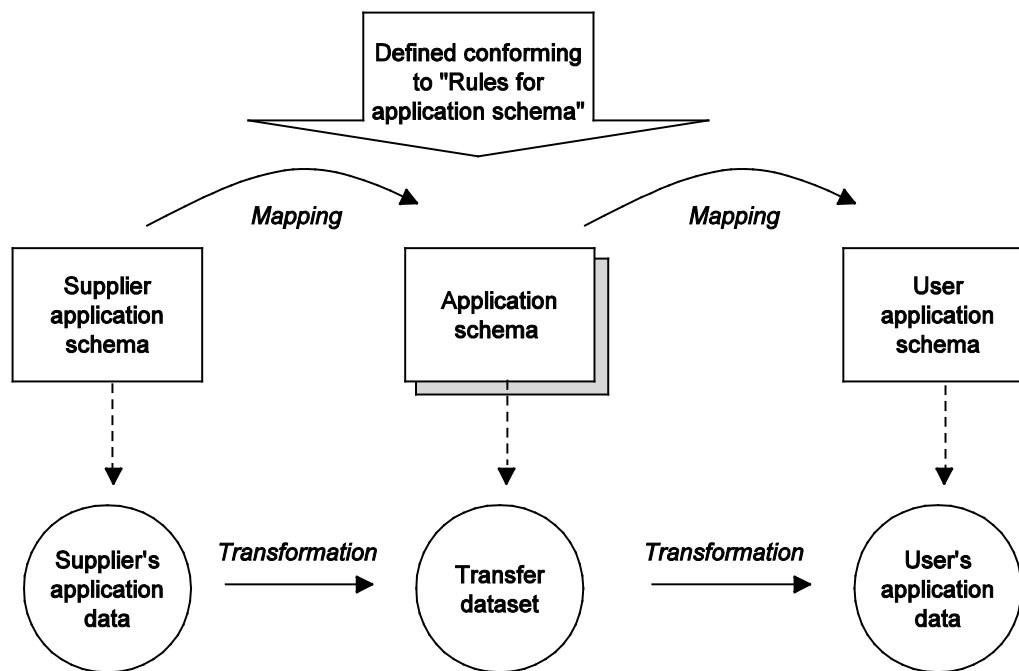


Figure 1 — Data interchange by transfer

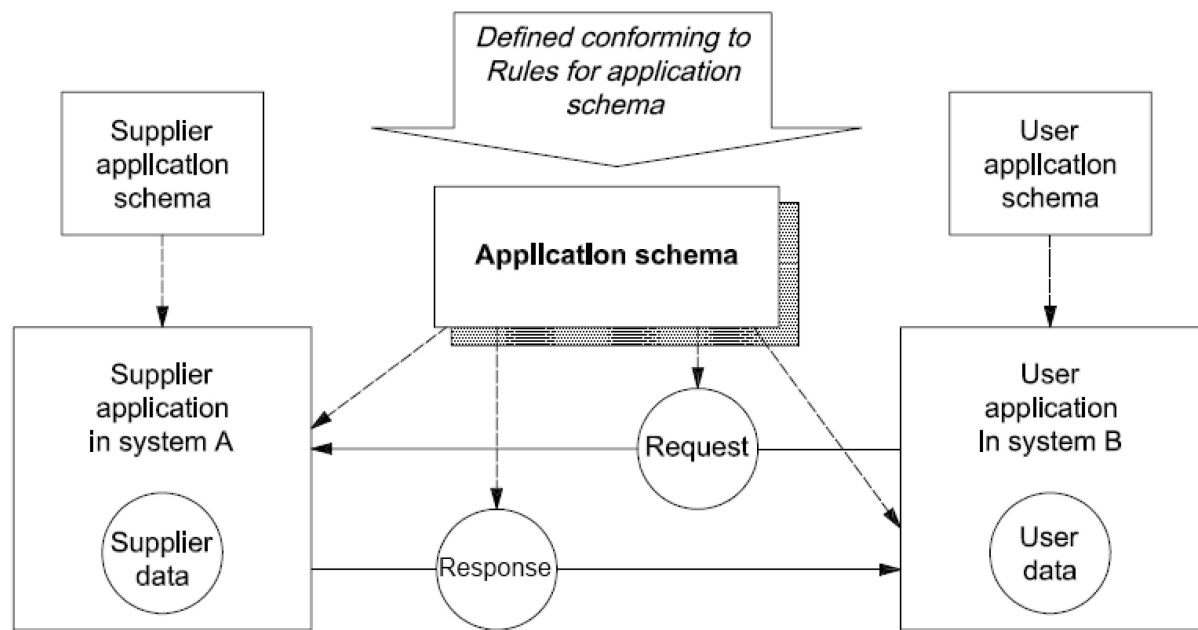
First, the user and the supplier must agree on creating an application schema for the data being exchanged in accordance with this International Standard. In order to facilitate the transfer of data, the application schema is developed using the application schemas from the user and the supplier. One mapping will be made from the supplier's application schema to this application schema for the exchanged data, and a second mapping will be made from this application schema to the application schema of the user.

Second, the supplier must be able to transform 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 must 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.

6.3.3 Data interchange by transactions

Figure 2 shows data interchange through transactions described in the interoperability model. The user application makes a request for data that is received by the supplier application. In response, the supplier application delivers a resulting dataset. Both the request and the resulting dataset are defined according to a common application schema. The supplier application is responsible for transforming the data in system A into the data in the exchanged dataset. After receipt, the user application is responsible for transforming the exchanged data into data in system B. Data interchange by transaction is provided by geospatial services as defined in ISO 19119:2005. In particular, feature access services are defined in Geographic model/information management services.



NOTE The unbroken lines show the flow of data. Broken lines denote the role of the application schema on the data interchange.

Figure 2 — Data interchange by transactions

7 Principles for defining features

7.1 General

This clause describes general rules for formalizing an application schema, independent of any specific conceptual schema language. The General Feature Model is the meta-model for application schemas, and the definition of the General Feature Model is part of the general rules. The rules comprise a requirements class summarized in Table 15.

Table 15 — Requirements class covering general requirements and the meta-model for application schemas

Requirements class	/req/general
Target type	Application schema
Requirement	/req/general/csl
Requirement	/req/general/integration
Requirement	/req/general/feature
Requirement	/req/general/property-name
Requirement	/req/general/attribute
Requirement	/req/general/operation
Requirement	/req/general/association-role
Requirement	/req/general/value-assignment
Requirement	/req/general/feature-association
Requirement	/req/general/inheritance
Recommendation	/req/general/constraint

7.2 Features, Coverages and Properties

7.2.1 Features

A fundamental unit of geographic information is called a feature. ISO 19110:2005 provides a standard framework for organizing and reporting the classification of features.

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

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 occurrence.

Figure 3 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.

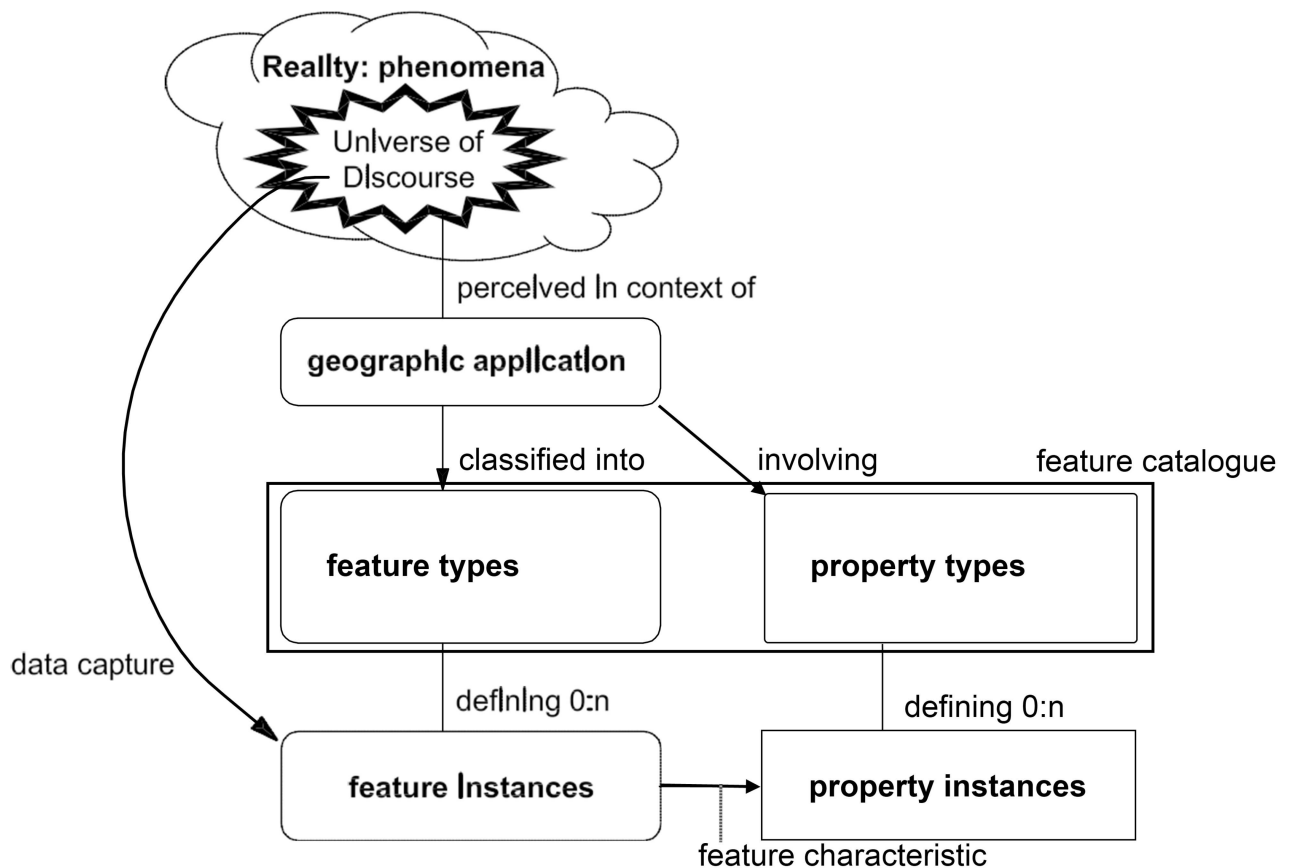


Figure 3 — The process from universe of discourse to data

7.2.2 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 focussing 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 may be used to infer the existence of specific objects or features on the ground.

EXAMPLE 2 Signals in a geophysical borehole log may 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 may 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 may 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 (7.5.8).

While the coverage model is described in detail in ISO 19123:2005, 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.

7.2.3 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 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 an estimate, with an error resulting from the observation procedure.

NOTE Following ISO 19156:2011, 'observation' denotes the generalized case of an event using a procedure to generate the estimate. The procedure may involve a sensor, but may 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.

7.3 Features and the application schema

This International Standard supports the definition of feature types and coverages with respect to their representation in data structures defined by application schemas.

Figure 4 shows the process of structuring data from the universe of discourse to the geographic dataset. The definitions of the feature types and their properties, as perceived in context of an application field, will be derived from the universe of discourse. A feature catalogue documents the feature types.

An application schema defines the logical structure of data and may define operations that can be performed on or with the data. An application schema addresses the logical organization, rather than the physical. An application schema is formalized using a Conceptual Schema Language.

/req/general/csl	An application schema shall be expressed in a conceptual schema language. It shall describe the structure and content of the dataset that represents a universe of discourse.
-------------------------	---

When creating an application schema, the concepts of the General Feature Model (GFM; see 7.4) are mapped to the concepts of the chosen conceptual schema language. For UML, these rules are described in Clause 8.

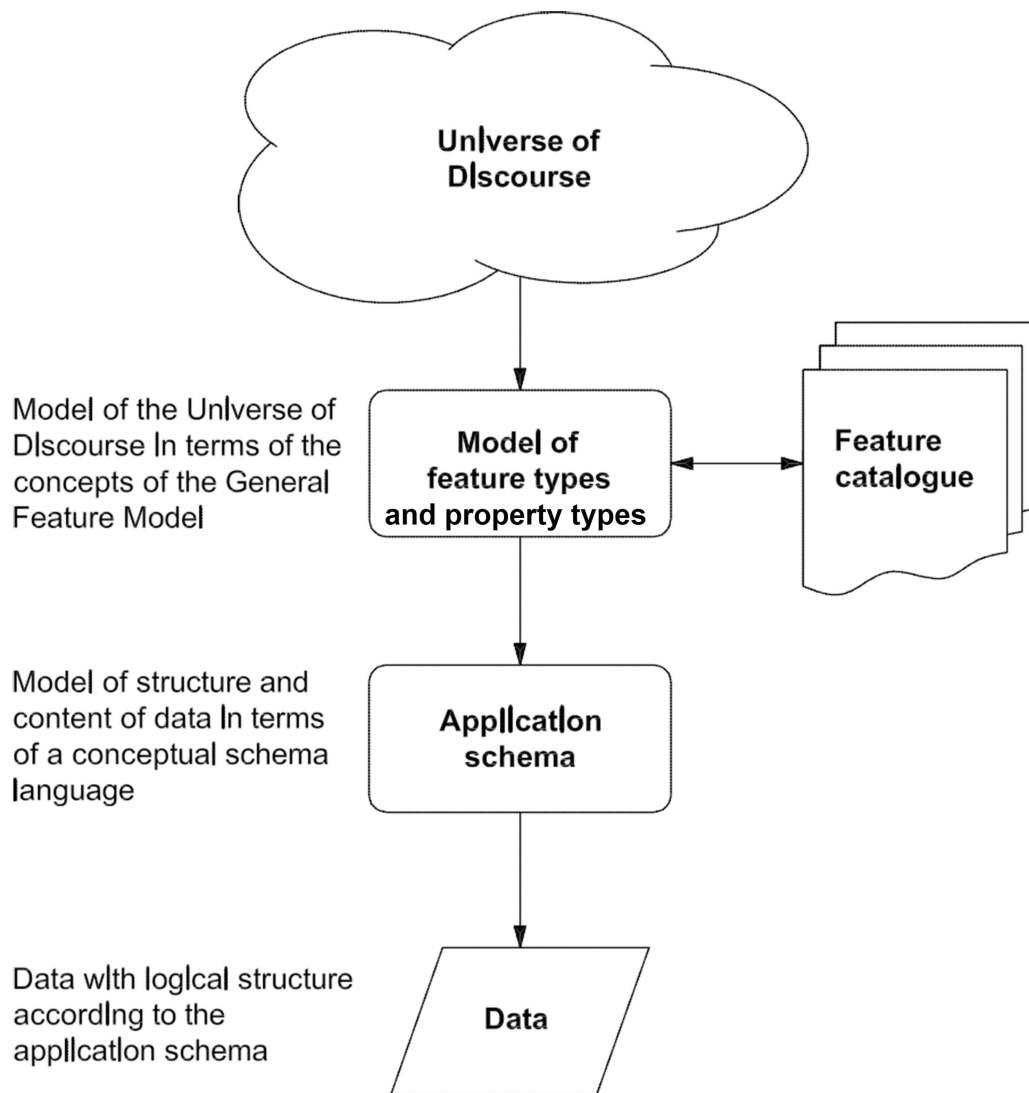


Figure 4 — From reality to geographic data

Use of definitions from existing feature catalogues can reduce the costs of data acquisition, allowing developers to use existing data, and simplify the process of developing the application schema. Dependencies are clearly recorded.

/req/general/integration

The developer of an application schema may use definitions from feature catalogues that already exist. Where an application schema integrates components from another schema, the dependencies shall be recorded explicitly using mechanisms provided by the conceptual schema language.

7.4 The General Feature Model

7.4.1 Introduction

Concepts used to define features and how these concepts are related are identified and described in 7.4. The description is expressed in a conceptual model, also called the General Feature Model (GFM).

Annex B provides discussions regarding the purpose and design of the GFM.

NOTE The complete kernel of GFM is found in Annex B.3.

7.5 to 7.7 depict how different aspects of the properties of a feature are managed. 7.5 describes different aspects of attributes and 7.6 describes different aspects of relationships. 7.7 describes the concept of constraints.

7.4.2 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 a CSL, which is UML in accordance with ISO 19103:201X.

The things we want to classify are called features. Feature types have properties that are feature attributes, feature operations 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 Model (FCM) that realizes the GFM concepts and adds more concepts (see ISO 19110:2005). 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 standard schemas. As we want to integrate standard schemas into our application schemas, it is convenient to also express the application schema in UML. This International Standard defines the main rules for mapping GFM concepts into UML. This may be done for other CSLs as well. As UML provides a general mechanism for classifications of any kind, it has to be profiled to be used as a basis for the classification of features. This profile is UML as specified in ISO 19103:201X and augmented in this International Standard and is summarized in 8.2.2.

The GFM defines the structure for classifying features that we need to keep in mind when we make our application schema 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 (i.e. the UML profile as restricted by ISO 19103:201X and augmented in this International Standard) is the metamodel for an application schema.

7.4.3 The main structure 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,
- feature association roles characterizing the feature type, and
- defined behaviour of 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.

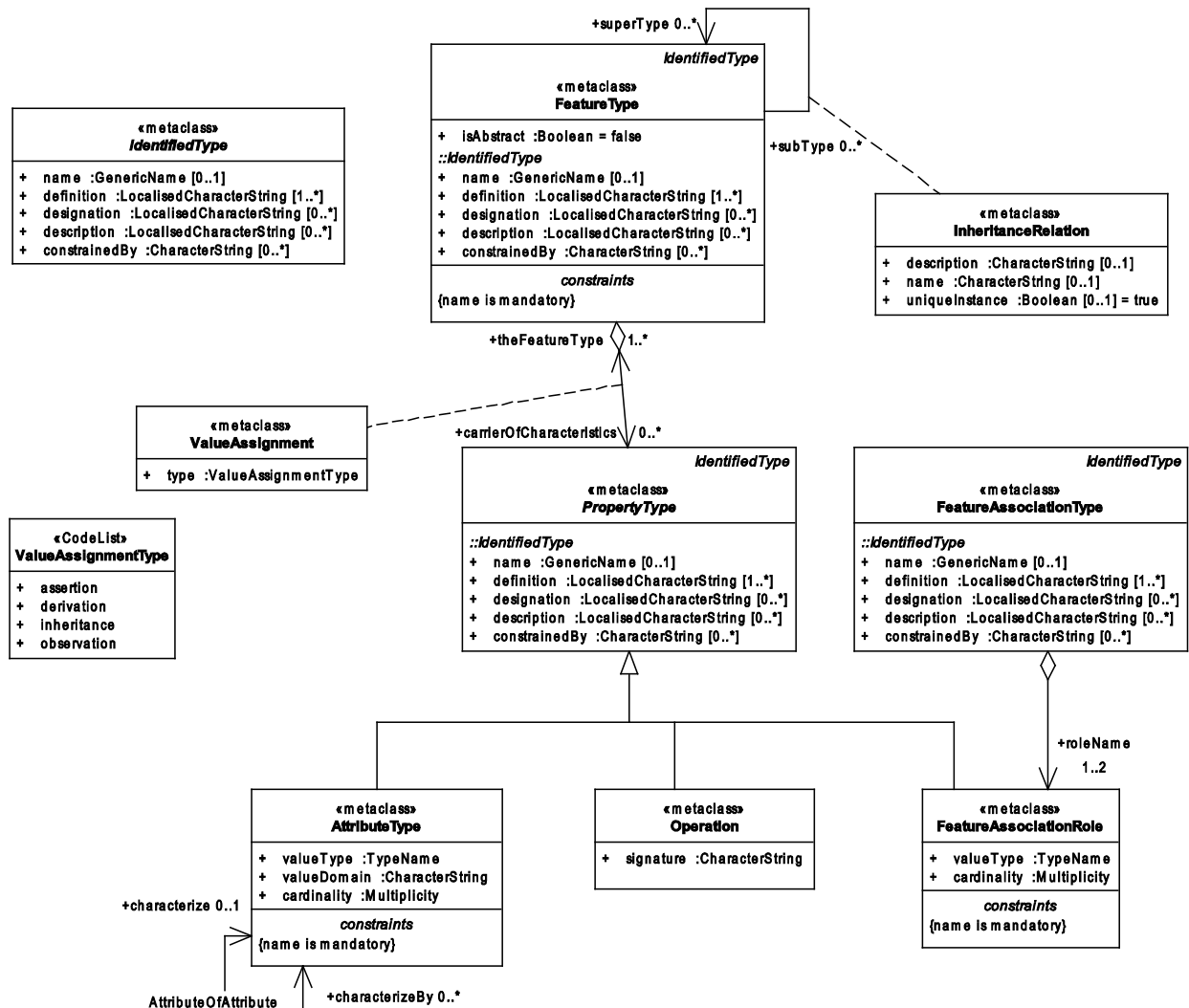


Figure 5 — The General Feature Model

NOTE 1 Earlier versions of this standard prefixed most class names with 'GF_'. That convention is no longer used.

NOTE 2 The types `CharacterString`, `LocalisedCharacterString`, `LocalName`, `TypeName`, `Multiplicity` are from ISO 19103:201X.

7.4.4 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`.

— name

name of the element. Optional in general, but required on some specializations as indicated below. The type `GenericName` [ISO 19103:201X] allows the namespace to be either explicit or implicit.

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] 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 is usually but not necessarily provided by the ontology. When using OWL as the CSL [ISO 19150-2:201X] 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**

concise definition of the element. One definition is mandatory. Additional definitions can be provided in multiple languages if required.

— **designation**

natural language designator for the element to complement the **name**. Optional, with multiple designations allowed to support different languages.

— **description**

description of the element, including information beyond that required for concise definition but which may assist in understanding its scope and application. Optional, with multiple descriptions allowed to support different languages.

7.4.5 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.

/req/general/feature	<p>Features shall be modelled as instances of the metaclass FeatureType, as defined in Figure 5 and in 7.4.5.</p> <p>Features shall not be modelled as specializations of GM_Object (ISO 19107:2003) or TM_Object (ISO 19108:2002).</p> <p>The feature shall have a name that is unique in the application schema.</p>
-----------------------------	--

NOTE 1 Feature types are equivalent to classes and feature instances are equivalent to objects, in object-oriented modelling.

NOTE 2 Geometry-, topology- and temporal-objects (GM_Object, TP_Object, TM_Object) are not abstractions of real-world phenomena. These types can provide types for feature properties, but cannot be specialized to features.

NOTE 3 Annex B provides a table on the use of the term “feature”.

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**

boolean attribute (mandatory). If true, the feature type acts as an abstract supertype.

— **carrierOfCharacteristics**

association role to specify any feature operation, 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).

7.4.6 PropertyType

PropertyType is the abstract superclass for AttributeType, Operation 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. PropertyType adds no further attributes or associations.

Property types are specified independent of any specific feature type, but a property instance and value occurs in the context of a feature. Nevertheless, the feature type or feature identity for an observed property may not be available, for example because the necessary data interpretation has not yet been done. Therefore, properties may be described independent of features. Property type names therefore need to also be unique within an application.

/req/general/property-name	The name of each property type shall be unique within an application schema.
-----------------------------------	--

7.4.7 AttributeType

AttributeType is the metaclass for attribute definitions of a feature type (see also 7.5).

/req/general/attribute	Feature attributes shall be modelled as instances of the metaclass AttributeType, as defined in Figure 5 and in 7.4.7, and further elaborated in 7.5.
-------------------------------	---

AttributeType inherits attributes and associations from the abstract metaclass PropertyType. The **name** is mandatory for an attribute type.

AttributeType adds the following attributes and association ends:

— **valueType**

data type of the attribute value (mandatory).

NOTE 1 ISO 19103:201X defines some data types that may be used for the valueType of a feature attribute, and other ISO geographic information standards define other object types that may be used.

EXAMPLE 1 Integer, character String or GM_Object.

— **valueDomain**

description of a set of values (mandatory). In some cases the domain of values may be a set of concepts used as classifiers, organized in a vocabulary or lexicon.

EXAMPLE 2 Positive, from 3 to 7; GM_Object and all its subtypes as defined in ISO 19107:2003; enumerated set.

NOTE 2 ISO 19103:201X provides for a class that defines a vocabulary to be stereotyped «Enumeration» or «CodeList», where the members of the vocabulary are modelled as UML attributes. Conventionally a vocabulary was considered as a set of character string values. However, modern knowledge organization theory considers a *term* to be merely a label for a *concept* (ISO/TS 19104:2008; Miles & Bechhofer, 2009).

— **cardinality**

number of instances of the attribute that may be associated with a single instance of a feature type (mandatory).

— **characterize**

the attribute type that is described by this attribute type (in the case of an attribute of attribute) (optional).

— **characterizeBy**

an attribute type that describes this attribute type (in the case of an attribute of attribute) (optional, repeatable).

EXAMPLE 3 An attribute that carries the position of a feature may have another attribute that holds the positional accuracy (data value of QualityAttributeType) of this position.

A taxonomy of attribute types is described in 7.5.

7.4.8 Operation

The behaviour of features is described by operations that may be performed upon or by all instances of a feature type. An Operation represents the behaviour of a feature type as a function or a method. ISO 19110:2005 provides a broader discussion on behaviour of feature types. Operation is the metaclass for describing behaviour of feature types in terms of operations.

/req/general/operation	Operations shall be modelled as instances of the metaclass Operation, as defined in Figure 5 and in 7.4.8.
-------------------------------	--

NOTE 1 Operations only apply to the interoperability model and do not apply to the data transfer model as described in 6.3.

NOTE 2 Instances of Operation are of three kinds: observer operations, mutator operations and constructor operations. Observer operations return the current values of attributes. Mutator operations include actions that change those values. A constructor operation creates an instance of a class for which it is defined. For example, an observer operation may be used to find the height of a dam. Raising the dam is a mutator operation that changes the height of the dam and also affects the attributes of the watercourse and the reservoir associated with the dam. Values may be observed or affected for another feature instance if there is an association between the feature types involved.

In addition to the attributes and associations inherited from the abstract metaclass PropertyType, Operation adds the following attributes:

— **signature**

description that indicates the name, the arguments and the return values of an operation (mandatory).

NOTE 3 In UML, the signature is expressed in the form *operation_name(input_parameter1, input_parameter2,...) : output_value_type*, for example *has_height() : real*.

Properties of class operations are determined by application schema designers.

7.4.9 FeatureAssociationRole

FeatureAssociationRole is the metaclass for the classes of roles that are part of a FeatureAssociationType (7.4.11). 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.

/req/general/association-role	Feature association roles shall be modelled as instances of the metaclass <code>FeatureAssociationRole</code> , as defined in Figure 5 and in 7.4.9.
--------------------------------------	--

`FeatureAssociationRole` inherits attributes and associations from the abstract metaclass `PropertyType`. The **name** is mandatory for an association role.

`FeatureAssociationRole` adds the following attributes:

— **valueType**

data type of the role value (mandatory).

— **cardinality**

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).

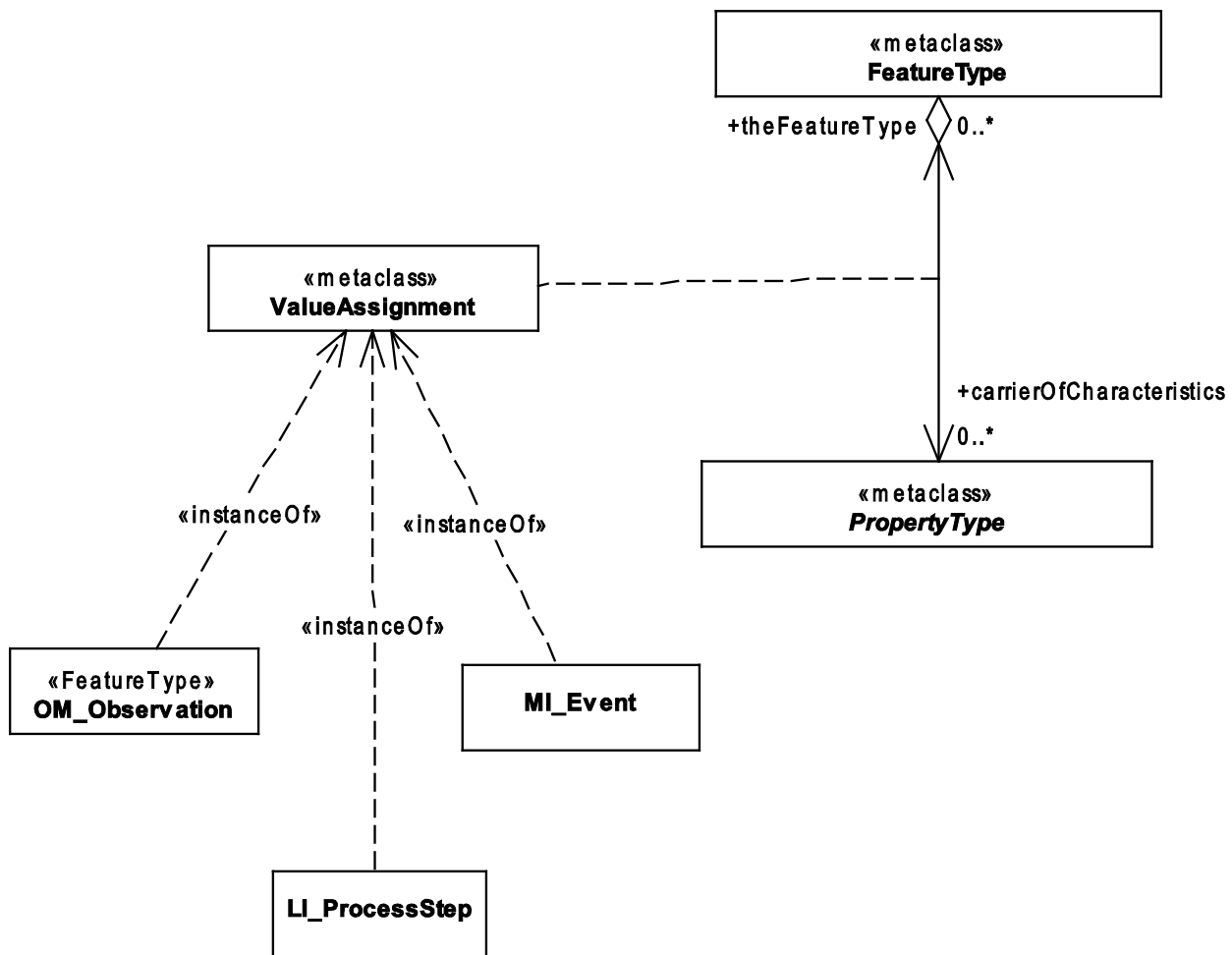
7.4.10 ValueAssignment

`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 `Operation`, `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.

/req/general/value-assignment	<p>A value-assignment process shall be modelled as an instance of the association class metaclass <code>ValueAssignment</code>, as defined in Figure 6 and in 7.4.10.</p> <p>In a value-assignment instance, the property for which a value is assigned shall be consistent with the feature-type that it is associated with.</p>
--------------------------------------	---

The feature type `OM_Observation` (ISO 19156:2011), and the classes `LI_ProcessStep` (ISO 19115-1:201X) and `MI_Event` (ISO 19115-2:2009) 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 OM_Observation class in ISO 19156:2011 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.

EXAMPLE 1 The owner of a land parcel or owner of a building is typically asserted by a land registration authority.

— Inheritance from a parent feature or type in which a property value is duplicated from a well defined source.

EXAMPLE 2 All instances of 'daffodil' have the property colour='yellow' inherited from their class definition.

— Derivation, in which a property value is derived from a number of input parameters based on a defined ruleset.

EXAMPLE 3 If a land area feature has a surface property that equals 'paved', a type property that equals 'road or track' and a make property that equals 'manmade', then the classification value 'road' is assigned.

EXAMPLE 4 Spatial analysis: The municipality a feature falls into given by a point location is automatically determined by spatial analysis against a municipal area dataset.