International Standard

ISO 19103

Geographic information — Conceptual schema language

Information géographique — Langage de schéma conceptuel

Second edition
2024-09

Reference number
ISO 19103:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 document 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 second edition cancels and replaces the first edition (ISO 19103:2015), which has been technically revised.

The main changes are as follows:

— conformance to UML 2.5.1 has been improved;

  — the UML profile has been improved and the stereotypes Leaf, CodeList and Union have been deprecated;

  — the collection data types, the name data types, the extension data types and data type Any have been removed;

— alignment with the data types described in ISO/IEC 11404:2007, Clause 8 and Clause 10 has been improved;

— the conformance classes for conceptual schemas modelled in UML 1.x and for conceptual schemas modelled in another conceptual schema language have been removed;

— the normative references have been updated, in particular:

  — addition of UML 2.5.1 and removal of ISO/IEC 19505-2:2012 (equivalent to UML 2.4.1, Superstructure[4]);

  — removal of the Object Constraint Language (OCL) specification.

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

This document is concerned with the adoption and use of a conceptual schema language (CSL) for developing computer-interpretable models, or schemas, of geographic information. Standardization of geographic information requires the use of a formal CSL to specify unambiguous schemas that can serve as a basis for data interchange. An important goal of the ISO 19100 family of documents is to create a framework in which data interchange and service interoperability can be realized across multiple implementation environments. The adoption and consistent use of a CSL to specify geographic information is of fundamental importance in achieving this goal.

There are two aspects to this document. First, a CSL is selected that meets the requirements for rigorous representation of geographic information. Several CSLs exist, of which two predominate in the geographic domain: the Unified Modeling Language (UML), specified by the Object Management Group (OMG), on the one hand, and the combination of the three Semantic Web specifications, the Resource Description Framework Schema (RDFS), the Web Ontology Language (OWL) and the Shapes Constraint Language (SHACL), specified by the World Wide Web Consortium (W3C), on the other hand. It was decided to continue using UML as it has proven its capability within the ISO 19100 family of documents, it supports a model-driven approach and it has a standardized graphical notation. This document identifies a subset of UML as the CSL for the specification of conceptual schemas. It also specifies a UML profile for the specification of conceptual schemas, and it specifies provisions on how to use UML and the UML profile to create conceptual schemas that are a basis for achieving the goal of interoperability. In addition, this document defines a set of core data type definitions for use in conceptual schemas.

One goal of the ISO 19100 family of documents using conceptual schemas specified in UML is that they will provide a basis for model-based mapping to encoding schemas like those defined in ISO 19118, as well as a basis for creating implementation specifications for implementation profiles for various other environments.

This document describes the general metamodel for the use of UML in the context of ISO geographic information documents. Aspects specifically dealing with the modelling of application schemas are described in ISO 19109.

In accordance with the ISO/IEC Directives, Part 2, 2021, *Principles and rules for the structure and drafting of ISO and IEC documents*, in International Standards the decimal sign is a comma on the line. However, the General Conference on Weights and Measures (*Conférence Générale des Poids et Mesures*) at its meeting in 2003 passed unanimously the following resolution: “The decimal marker shall be either a point on the line or a comma on the line.”[5] In practice, the choice between these alternatives depends on customary use in the language concerned. In the technical areas of geodesy and geographic information it is customary for the decimal point always to be used, for all languages. That practice is used throughout 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 — Conceptual schema language

1 Scope

This document specifies provisions for the use of a conceptual schema language within the context of modelling geographic information. The chosen conceptual schema language is a subset of the Unified Modeling Language (UML).

This document specifies a UML profile for modelling geographic information.

This document specifies a set of core data types for use in conceptual schemas.

The standardization target type of this document is conceptual schemas describing geographic information.

2 Normative references

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

UML 2.5.1: OBJECT MANAGEMENT GROUP (OMG). *Unified Modeling Language (UML)* [online]. Version 2.5.1. December 2017. Available at: https://www.omg.org/spec/UML/2.5.1

3 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
abstract
<information technology> filter out detail that is not within the scope of interest

Note 1 to entry: Abstracting facilitates the understanding of the essence of a *concept* (3.20) and allows for handling complexity.

Note 2 to entry: An act of abstracting is designated as an "abstraction". In the information technology domain, the term "abstraction" also represents concept *abstraction* (3.4).

3.2
abstract classifier
<UML> *classifier* (3.16) that has no direct *instances* (3.42)

Note 1 to entry: UML 2.5.1, 9.2.3.2 requires that every instance of an abstract classifier is an instance of one of its specializations.

Note 2 to entry: Adapted from UML 2.5.1, 9.2.3.2.

3.3
abstract schema
conceptual schema (3.23) that is not implementable without further specification

EXAMPLE The conceptual schemas for describing the spatial characteristics of geographic entities defined in ISO 19107:2019.

Note 1 to entry: An abstract schema can be applied to many domains.

Note 2 to entry: An abstract schema can be realized by an *application schema* (3.8).

3.4
abstraction
<information technology> result of an act of *abstracting* (3.1)

3.5
abstraction
<UML> *dependency* (3.30) that relates two *named elements* (3.50) or sets of named elements that represent the same *concept* (3.20) at different *levels of abstraction* (3.45) or from different viewpoints

Note 1 to entry: Adapted from UML 2.5.1, 7.7.3.3.

3.6
aggregation
shared aggregation
<UML> *binary association* (3.12) that specifies a *part-whole relation* (3.58) where the whole does not have responsibility for the existence of its parts

Note 1 to entry: A part can be included in more than one whole simultaneously.

Note 2 to entry: Adapted from UML 2.5.1, 9.5.3.

3.7
application
manipulation and processing of data in support of user requirements

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

3.8
application schema
conceptual schema (3.23) for data required by one or more *applications* (3.7)

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

3.9
association
<UML> semantic *relationship* (3.63) that can occur between *instances* (3.42) that have a *type* (3.70)

Note 1 to entry: An association is also a kind of *classifier* (3.16).

Note 2 to entry: Adapted from UML 2.5.1, 11.5.3.1.

3.10
attribute
<UML> *property* (3.61) owned by a *classifier* (3.16) other than an *association* (3.9)

Note 1 to entry: Adapted from UML 2.5.1, 9.5.3.

3.11
behavioural feature
<UML> *feature* (3.36) that specifies an aspect of behaviour

Note 1 to entry: Adapted from UML 2.5.1, 9.9.2.1.

3.12
binary association
<UML> *association* (3.9) having two member ends

Note 1 to entry: UML 2.5.1, 11.5.3.1 permits that the two member ends have the same *type* (3.70).

Note 2 to entry: Adapted from UML 2.5.1, 11.5.3.1.

3.13
cardinality
<UML> number of values

EXAMPLE The cardinality of a collection having three values is three.

Note 1 to entry: Cardinality is a characteristic of a collection.

Note 2 to entry: Adapted from UML 2.5.1, 7.5.3.2.

3.14
class
<UML> *classifier* (3.16) of a set of *objects* (3.54)

Note 1 to entry: Adapted from UML 2.5.1, 11.8.3.1.

3.15
class diagram
<UML> structure diagram where the primary symbols in the contents area are either *class* (3.14) symbols or *interface* (3.43) symbols, or both

Note 1 to entry: Adapted from UML 2.5.1, Annex A.

3.16
classifier
<UML> classification of *instances* (3.42) according to their *features* (3.36)

Note 1 to entry: A classifier is a kind of *type* (3.70).

Note 2 to entry: Adapted from UML 2.5.1, 9.2.1.

3.17
code set
code element set
code
code list
result of applying a coding scheme to all elements of a coded set

EXAMPLE The three-letter representations of airport names.

Note 1 to entry: The term “code” also represents the concept defined in ISO 19118:2011, 4.3.

[SOURCE: ISO/IEC 2382:2015, 2121556, modified — An additional admitted term “code list” has been added, the definition has been adjusted to use the terms used in Annex H, Note 1 to entry has been converted into an Example, Notes 2 and 3 to entry have been removed and a new Note 1 to entry has been added.]

3.18
comment
note
<UML> textual annotation that can be attached to a set of elements

Note 1 to entry: Adapted from UML 2.5.1, 7.8.2.1.

3.19
composition
<UML> *binary association* (3.12) that specifies a *part-whole relation* (3.58) where the whole has responsibility for the existence of its parts

Note 1 to entry: A part can only be included in at most one whole at a time.

Note 2 to entry: Adapted from UML 2.5.1, 9.5.3.

3.20
concept
unit of knowledge created by a unique combination of characteristics

Note 1 to entry: Concepts are not necessarily bound to particular *natural languages* (3.52). They are, however, influenced by social or cultural background, which often leads to different categorizations.

Note 2 to entry: This is the concept "concept" as used and designated by the term "concept" in terminology work. It is a very different concept from that designated by other domains such as industrial automation or marketing.

[SOURCE: ISO 1087:2019, 3.2.7]

3.21
conceptual formalism
set of modelling *concepts* (3.20) used to describe a *conceptual model* (3.22)

EXAMPLE Object-oriented modelling.

Note 1 to entry: One conceptual formalism can be expressed in several *conceptual schema languages* (3.24).

[SOURCE: ISO 19101-1:2014, 4.1.4, modified — Examples have been replaced.]

3.22
conceptual model
model (3.48) that defines *concepts* (3.20) of a *universe of discourse* (3.72)

Note 1 to entry: A model can include relations between concepts. A relation is a concept too.

[SOURCE: ISO 19101-1:2014, 4.1.5, modified — Note to entry added.]

3.23
conceptual schema
formal description of a *conceptual model* (3.22)

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

3.24
conceptual schema language
formal language (3.37) based on a *conceptual formalism* (3.21) for the purpose of representing *conceptual schemas* (3.23)

EXAMPLE UML, EXPRESS, IDEF1X.

Note 1 to entry: A conceptual schema language can be lexical or graphical. Several conceptual schema languages can be based on the same conceptual formalism.

[SOURCE: ISO 19101-1:2014, 4.1.7]

3.25
constraint
<UML> condition or restriction expressed in *natural language* (3.52) text or in a machine readable language for the purpose of declaring some of the semantics of an element or set of elements

Note 1 to entry: Adapted from UML 2.5.1, 7.8.3.1.

3.26
data type
set of distinct values, characterized by properties of those values, and by *operations* (3.55) on those values

EXAMPLE The data type "Boolean" with properties "unordered", "exact" and "non-numeric", and with operations "equal", "not", "and" and "or".

Note 1 to entry: Properties of data type values are ordered or unordered, exact or approximate, numeric or non-numeric and, if ordered, bounded or unbounded, as described in ISO/IEC 11404.

[SOURCE: ISO/IEC 11404:2007, 3.12, modified — Note to entry and example added.]

3.27
data type
<UML> *classifier* (3.16) whose *instances* (3.42) are distinguished only by their value

Note 1 to entry: Adapted from UML 2.5.1, 10.2.1.

3.28
data value
<UML> *instance* (3.42) of a *data type* (3.27)

Note 1 to entry: Adapted from UML 2.5.1, 7.5.3.2.

3.29
definition
representation of a *concept* (3.20) by an expression that describes it and differentiates it from related concepts

[SOURCE: ISO 1087:2019, 3.3.1]

3.30
dependency
<UML> *directed relationship* (3.32) which signifies that a single model element or a set of model elements requires other model elements for their specification or implementation

Note 1 to entry: A dependency signifies a supplier/client *relationship* (3.63) between model elements where the modification of a supplier can impact the client model elements. The complete semantics of the client element(s) are either semantically or structurally dependent on the *definition* (3.29) of the supplier element(s).

Note 2 to entry: Adapted from UML 2.5.1, 7.7.1 and 7.8.4.

3.31
designation
designator
label
representation of a *concept* (3.20) by a sign which denotes it in a domain or subject

Note 1 to entry: A designation can be linguistic or non-linguistic. It can consist of various types of characters, but also punctuation marks such as hyphens and parentheses, governed by domain-, subject-, or language-specific conventions.

Note 2 to entry: A designation can be a term including appellations, a proper name, or a symbol.

[SOURCE: ISO 1087:2019, 3.4.1, modified — An additional admitted term "label" has been added.]

3.32
directed relationship
<UML> *relationship* (3.63) between a collection of source model elements and a collection of target model elements

Note 1 to entry: Adapted from UML 2.5.1, 7.8.5.1.

3.33
enumeration
<UML> *data type* (3.27) whose values are named individually in the *model* (3.48)

Note 1 to entry: The set of *enumeration literals* (3.34) owned by an enumeration is ordered.

Note 2 to entry: Adapted from UML 2.5.1, 10.5.3

3.34
enumeration literal
<UML> user-defined *data value* (3.28) for an *enumeration* (3.33)

Note 1 to entry: In this case, the user is the modeller.

Note 2 to entry: Adapted from UML 2.5.1, 10.5.4.1.

3.35
extension
<UML> *association* (3.9) which indicates that the *properties* (3.61) of a *metaclass* (3.46) are extended through a *stereotype* (3.65)

Note 1 to entry: Adapted from UML 2.5.1, 12.4.1.1.

3.36
feature
<UML> characteristic

Note 1 to entry: Adapted from UML 2.5.1, 9.4.3.1.

3.37
formal language
language that is machine readable and has well-defined semantics

Note 1 to entry: Well-defined semantics will typically be model-theoretic semantics.

[SOURCE: ISO/IEC 21838-1:2021, 3.10]

3.38
generalization
<UML> taxonomic *directed relationship* (3.32) between a more general *classifier* (3.16) and a more specific classifier

Note 1 to entry: The more general classifier is called the parent, or the superclass if the classifier is a *class* (3.14). The more specific classifier is called the child. The generalization is directed from the child to the parent. The classifiers that can be reached by following the generalizations from a given classifier in the direction towards the more general classifiers are called the classifier's generalizations. The classifiers that can be reached by following the generalizations from a given classifier in the direction towards the more specific classifiers are called the classifier's specializations.

Note 2 to entry: Each *instance* (3.42) of the specific classifier is also an instance of the general classifier. The specific classifier inherits the *features* (3.36) of the more general classifier.

Note 3 to entry: Adapted from UML 2.5.1, 9.2.3.2 and 9.9.7.

3.39
identifier
linguistically independent sequence of characters capable of uniquely and permanently identifying that with which it is associated

[SOURCE: ISO 19135-1:2015, 4.1.5]

3.40
identity
inherent characteristic of an *instance* (3.42) that distinguishes it from all other instances

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1865, modified — The term "property" has been replaced by "characteristic" in the definition, and the Note to entry has been removed.]

3.41
inheritance
<information technology> mechanism by which more specific entities incorporate structure and behaviour defined by more general entities

3.42
instance
individual entity

Note 1 to entry: The term "instance" represents the same concept as the term "particular" defined in ISO/IEC 21838-1:2021, 3.3.

3.43
interface
<UML> *classifier* (3.16) that represents a declaration of a set of coherent public *features* (3.36) and obligations that together constitute a coherent service

Note 1 to entry: An interface specifies a contract; UML 2.5.1, 10.4.3.1 requires that any *instance* (3.42) of a classifier that realizes the interface fulfils that contract. The obligations associated with an interface are in the form of *constraints* (3.25) (such as pre- and post-conditions) or protocol specifications, which can impose ordering restrictions on interactions through the interface.

Note 2 to entry: Adapted from UML 2.5.1, 10.4.3.1.

3.44
keyword
<UML> reserved word that is an integral part of the UML notation

Note 1 to entry: Keywords normally appear as text annotations attached to a UML graphic element or as part of a text line in a UML diagram. Keywords are enclosed in guillemets («keyword») and thus have the same notation as *stereotyped* (3.65) model elements.

Note 2 to entry: Adapted from UML 2.5.1, Annex C.

3.45
level of abstraction
abstraction level
indication of the amount of detail that is outside the scope of interest

Note 1 to entry: A *model* (3.48) at a high level of abstraction has a relatively low amount of detail.

3.46
metaclass
<UML> *class* (3.14) in a *metamodel* (3.47)

EXAMPLE The class "Interface" is a class in the UML metamodel and is therefore a metaclass.

Note 1 to entry: Adapted from UML 2.5.1, 6.2.

3.47
metamodel
model (3.48) that defines a modelling language

EXAMPLE The UML metamodel.

Note 1 to entry: A model is an *instance* (3.42) of a metamodel, and a metamodel is an instance of a meta-metamodel.

Note 2 to entry: Adapted from the MDA Guide.[15]

3.48
model
abstraction (3.4) of some aspects of reality

[SOURCE: ISO 19109:2015, 4.15]

3.49
multiplicity
<UML> specification of the valid *cardinalities* (3.13)

EXAMPLE An *instance* (3.42) of a collection specified as having a multiplicity of "1..3" has at least one value and has not more than three values.

Note 1 to entry: A multiplicity is a definition of an inclusive interval of non-negative integers beginning with a lower bound and ending with a (possibly infinite) upper bound.

Note 2 to entry: Adapted from UML 2.5.1, 7.5.3.2 and 7.8.8.1.

3.50
named element
<UML> element in a *model* (3.48) that can have a name

Note 1 to entry: Adapted from UML 2.5.1, 7.8.9.

3.51
namespace
<UML> *named element* (3.50) that either owns or imports, or both, a set of named elements that can be identified by a name

Note 1 to entry: Adapted from UML 2.5.1, 7.8.10.1.

3.52
natural language
language which is or was in active use in a community of people, and the rules of which are mainly deduced from usage

[SOURCE: ISO 5127:2017, 3.1.5.02, modified — The Note to entry has been removed.]

3.53
n-ary association
<UML> *association* (3.9) having more than two member ends

Note 1 to entry: An association with three members ends is called a ternary association.

Note 2 to entry: Adapted from UML 2.5.1., 11.5.3.1.

3.54
object
<UML> individual with a state and *relationships* (3.63) to other individuals

Note 1 to entry: An object is an *instance* (3.42) of a *class* (3.14).

Note 2 to entry: Adapted from UML 2.5.1, 6.3.1.

3.55
operation
<UML> *behavioural feature* (3.11) of an *interface* (3.43), *data type* (3.27) or *class* (3.14)

Note 1 to entry: UML 2.5.1, 9.6.3.1 permits that an operation is directly invoked on *instances* (3.42) of its featuring *classifiers* (3.16). The operation specifies the name, *type* (3.70), parameters and *constraints* (3.25) for such invocations.

Note 2 to entry: Adapted from UML 2.5.1, 9.6.3.1.