INTERNATIONAL STANDARD

ISO 19118

Second edition
2011-10-15

Geographic information — Encoding

Information géographique — Codage

Reference number
ISO 19118:2011(E)

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.

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 19118 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

This second edition cancels and replaces the first edition (ISO 19118:2005), which has been technically revised.

Introduction

This International Standard specifies the requirements for defining encoding rules used for interchange of geographic data within the set of International Standards known as the “ISO 19100 series”. An encoding rule allows geographic information defined by application schemas and standardized schemas to be coded into a system-independent data structure suitable for transport and storage. The encoding rule specifies the types of data being coded and the syntax, structure and coding schemes used in the resulting data structure. The resulting data structure can be stored on digital media or transferred using transfer protocols. It is intended that the data be read and interpreted by computers, but data can be in a form that is human readable.

The choice of one encoding rule for application-independent data interchange does not exclude application domains and individual nations from defining and using their own encoding rules that can be platform dependent or more effective with regard to data size or processing complexity. XML is a subset of ISO/IEC 8879 and has been chosen because it is independent of computing platform and interoperable with the World Wide Web.

This International Standard is divided into three logical sections. The requirements for creating encoding rules based on UML schemas are specified in Clauses 6 to 9. The requirements for creating encoding service are specified in Clause 10, and the requirements for XML-based encoding rules are specified in Annex A.

The XML-based encoding rule is intended for use as a neutral data interchange. It relies on the Extensible Markup Language (XML) and the ISO/IEC 10646 character set standards.

The geographic information standards are organized within the set of International Standards known as the “ISO 19100 series”. The background and the overall structure of this series of International Standards and the fundamental description techniques are defined in ISO 19101, ISO/TS 19103 and ISO/TS 19104.

Users of this International Standard can develop application schemas to formally describe geographic information. An application schema is compiled by integrating elements from other standardized conceptual schemas (e.g. ISO 19107). How this integration takes place is described in ISO 19109. The set of International Standards known as the “ISO 19100 series” also defines a set of common services that are available when developing geographic information applications. The common services are generally defined in ISO 19119 and cover access to, and processing of, geographic information according to the common information model. This International Standard covers implementation issues.

Geographic information — Encoding

1 Scope

This International Standard specifies the requirements for defining encoding rules for use for the interchange of data that conform to the geographic information in the set of International Standards known as the "ISO 19100 series".

This International Standard specifies

— requirements for creating encoding rules based on UML schemas,

— requirements for creating encoding services, and

— requirements for XML-based encoding rules for neutral interchange of data.

This International Standard does not specify any digital media, does not define any transfer services or transfer protocols, nor does it specify how to encode inline large images.

2 Conformance

2.1 Introduction

Two sets of conformance classes are defined for this International Standard.

2.2 Conformance classes related to encoding rules

All encoding rules shall pass all test cases of the abstract test suite in B.1. All encoding rules shall pass all test cases of the abstract test suite in B.2 and/or B.3.

Table 1 — Conformance classes related to encoding rules

Conformance class	Subclause of the abstract test suite
All encoding rules	B.1
Encoding rule with instance conversion	B.2
Encoding rule with schema conversion	B.3

2.3 Conformance classes related to encoding services

All encoding services shall pass all test cases of the abstract test suite in B.4. Depending on the capabilities of the encoding service, it shall pass all test cases of additional conformance classes in accordance with Table 2.

Table 2 — Conformance classes related to encoding services

Conformance class	Subclause of the abstract test suite
All encoding services	B.4
Generic encoding service	B.5
Service that encodes data	B.6
Service that decodes data	B.7
Service that generates an output data structure schema	B.8

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.

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 10646:2011, *Information technology — Universal Coded Character Set (UCS)*

ISO/TS 19103:2005, *Geographic information — Conceptual schema language*

ISO 19109:2005, *Geographic information — Rules for application schema*

Extensible Markup Language (XML) 1.0, W3C Recommendation. Available at <http://www.w3.org/TR/REC-xml>

4 Terms and definitions

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

4.1
application schema
conceptual schema (4.5) for **data** (4.8) required by one or more applications

[ISO 19101:2002, 4.2]

NOTE An application schema describes the content, the structure and the constraints applicable to **information** (4.22) in a specific application domain.

4.2
character
member of a set of elements that is used for the representation, organization, or control of **data** (4.8)

[ISO/IEC 2382-1:1993, 01.02.11]

4.3
code
representation of a label according to a specified scheme

4.4
conceptual model
model (4.27) that defines concepts of a **universe of discourse** (4.33)

[ISO 19101:2002, 4.4]

4.5
conceptual schema
formal description of a **conceptual model** (4.4)

[ISO 19101:2002, 4.5]

4.6
conceptual schema language
formal language based on a conceptual formalism for the purpose of representing **conceptual schemas** (4.5)

[ISO 19101:2002, 4.6]

EXAMPLES UML, EXPRESS, IDEF1X.

NOTE A conceptual schema language may be lexical or graphical.

4.7
conversion rule
rule for converting instances in the input **data** (4.8) structure to instances in the output data structure

4.8
data
reinterpretable representation of **information** (4.22) in a formalized manner suitable for communication, interpretation, or processing

[ISO/IEC 2382-1:1993, 01.01.02]

4.9
data interchange
delivery, receipt and interpretation of **data** (4.8)

4.10
data transfer
movement of **data** (4.8) from one point to another over a **medium** (4.26)

NOTE Transfer of **information** (4.22) implies transfer of data.

4.11
data type
specification of a **value domain** (4.34) with operations allowed on values in this domain

[ISO/TS 19103:2005, 4.1.5]

EXAMPLES Integer, Real, Boolean, String and Date.

NOTE A data type is identified by a term, e.g. Integer. Values of the data types are of the specified value domain, e.g. all integer numbers between –65537 and 65536. The set of operations can be +, -, * and / and is semantically well defined. A data type can be simple or complex. A simple data type defines a value domain where values are considered atomic in a certain context, e.g. Integer. A complex data type is a collection of data types that are grouped together. A complex data type may represent an object and can, thus, have identity.

4.12
dataset
identifiable collection of **data** (4.8)

[ISO 19115:2003, 4.2]

4.13
encoding
conversion of **data** (4.8) into a series of **codes** (4.3)

4.14
encoding rule
identifiable collection of **conversion rules** (4.7) that define the **encoding** (4.13) for a particular **data** (4.8) structure

EXAMPLES XML, ISO 10303-21, ISO/IEC 8211.

NOTE An encoding rule specifies the types of data being converted as well as the syntax, structure and **codes** (4.3) used in the resulting data structure.

4.15
encoding service
software component that has an **encoding rule** (4.14) implemented

4.16
feature
abstraction of real world phenomena

[ISO 19101:2002, 4.11]

NOTE A feature may occur as a type or an instance. Feature type or feature instance is used when only one is meant.

4.17
file
named set of records stored or processed as a unit

[ISO/IEC 2382-1:1993, 01.08.06]

4.18
geographic data
data (4.8) with implicit or explicit reference to a location relative to the Earth

[ISO 19109:2005, 4.12]

4.19
geographic information
information (4.22) concerning phenomena implicitly or explicitly associated with a location relative to the Earth

[ISO 19101:2002, 4.16]

4.20
identifier
linguistically independent sequence of **characters** (4.2) capable of uniquely and permanently identifying that with which it is associated

[ISO 19135:2005, 4.1.5]

4.21
identification convention
set of rules for creating **identifiers** (4.20)

4.22
information
knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning

[ISO/IEC 2382-1:1993, 01.01.01]

4.23
instance model
representation **model** (4.27) for storing **data** (4.8) according to an **application schema** (4.1)

4.24
interface
⟨UML⟩ named set of operations that characterize the behaviour of an element

[ISO/IEC 19501]

4.25
interoperability
capability to communicate, execute programs, or transfer **data** (4.8) among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

[ISO/IEC 2382-1:1993, 01.01.47]

4.26
medium
substance or agency for storing or transmitting **data** (4.8)

EXAMPLES Compact disc, internet[1], radio waves, etc.

4.27
model
abstraction of some aspects of reality

[ISO 19109:2005, 4.14]

4.28
schema
formal description of a **model** (4.27)

[ISO 19101:2002, 4.25]

4.29
schema model
representation **model** (4.27) for storing **schemas** (4.28)

EXAMPLE Representation model for a schema repository.

4.30
stereotype
⟨UML⟩ new type of modelling element that extends the semantics of the metamodel

[ISO/IEC 19501]

NOTE It is necessary that stereotypes be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure, of pre-existing types and classes. Certain stereotypes are predefined in the UML, others may be user-defined. Stereotypes are one of three extensibility mechanisms in UML; the others are constraint and tagged value.

4.31
transfer protocol
common set of rules for defining interactions between distributed systems

4.32
transfer unit
collection of **data** (4.8) for the purpose of a **data transfer** (4.10)

NOTE A transfer unit does not have to be identifiable like a **dataset** (4.12).

4.33
universe of discourse
view of the real or hypothetical world that includes everything of interest

[ISO 19101:2002, 4.29]

4.34
value domain
set of accepted values

[ISO/TS 19103:2005, 4.1.15]

EXAMPLE The range 3-28, all integers, any character, enumeration of all accepted values (green, blue, white).

5 Symbols and abbreviated terms

DCE Distributed computing environment

DUID Domain unique identifier

HTML Hypertext markup language

MODIS Moderate resolution imaging spectroradiometer

POSC Petroleum Open Standards Consortium

TIFF Tagged image file format

UCS Universal multiple-octet coded character set

UML Unified modelling language

UTF UCS Transfer format

UUID Universally unique identifier

XML Extensible markup language

6 Fundamental concepts and assumptions

6.1 Concepts

The purpose of the set of International Standards known as the “ISO 19100 series” is to enable interoperability between heterogeneous geographic information systems. To achieve interoperability between heterogeneous systems, it is necessary to determine two fundamental issues. The first issue is to define the semantics of the content and the logical structures of geographic data. This shall be done in an application schema. The second issue is to define a system- and platform-independent data structure that can represent data corresponding to the application schema.

The fundamental concepts of data interchange, i.e. the procedure based on the application schema for encoding, delivery, receipt and interpretation of geographic data, are described in 6.2 to 6.6. An overview of the data interchange process is described in 6.2; 6.3 introduces application schemas that allow interpretation of geographic data; 6.4 describes the importance of the encoding rule for producing system-independent data structures; 6.5 describes a software component, called the encoding service, for executing the encoding rule; and 6.6 describes the procedure for delivery and receipt, called the transfer service.

6.2 Data interchange

An overview of a data interchange is shown in Figure 1. System A wants to send a dataset to system B. To ensure a successful interchange, it is necessary that A and B decide on three things: i.e. a common application schema **I**, which encoding rule *R* to apply, and what kind of transfer protocol to use. The application schema is the basis of a successful data transfer and defines the possible content and structure of the transferred data, whereas the encoding rule defines the conversion rules for how to code the data into a system-independent data structure.

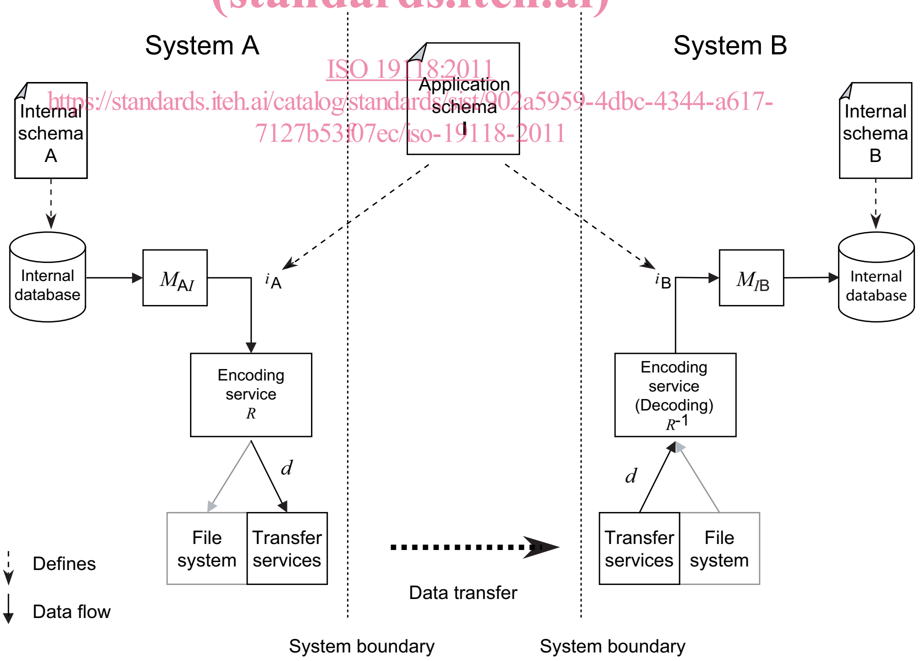

Figure 1 — Overview of data interchange between two systems

Both systems, A and B, store data in an internal database according to an internal schema, but the schemas are usually different, i.e. schema A is not equal to schema B. It is necessary to take the following logical steps in order to transfer a dataset from A's internal database to B's internal database.

a) The first step for system A is to translate its internal data into a data structure that is in accordance with the common application schema I. Here, this is done by defining a mapping from the concepts of the internal schema to the concepts defined in the application schema and by writing appropriate mapping software to translate the data instances. In Figure 1, this mapping is denoted as $M_{\mathrm{A}I}$. The result is an application-schema-specific data structure, i_{A}. The data structure is stored in memory or on an intermediate file and is system-dependent and, thus, is not suitable for transfer.

b) The next step is to use an encoding service, which applies the encoding rule R to create a data structure that is system independent and, therefore, suitable for transfer. This encoded dataset is called d and may be stored in a file system or transferred using a transfer service.

c) System A then invokes a transfer service to send the encoded dataset d to system B. The transfer service follows a transfer protocol for how to do packaging and how the actual transportation over an on-line or off-line communication medium should take place. It is necessary that both parties agree on the transfer protocol used.

d) The transfer service on system B receives the transferred dataset, and according to the protocol the dataset is unpacked and stored as an encoded dataset d, e.g. on an intermediate file.

e) In order to get an application-schema-specific data structure i_{B}, system B applies the inverse encoding rule R^{-1} to interpret the encoded data.

f) To use the dataset, it is necessary that B translate the application-schema-specific data structure i_{B} into its internal database. This is done by defining a mapping from the application schema into its internal schema and by writing software that does the actual translation. In Figure 1 this mapping is denoted $M_{I\mathrm{B}}$.

This International Standard specifies only the requirements for creating encoding rules and the encoding services and not the whole data interchange process. Thus, only steps b) and e) are standardized. Steps a), c), d) and f) use general information technology services.

6.3 Application schema

An application schema is a conceptual schema for applications with similar data requirements. The application schema is the basis of a successful data interchange and defines the possible content and structure of the data. It is also the basis for implementing application-schema-specific data structures for local storage of data.

The application schema used for encoding in compliance with this International Standard shall be written in the UML conceptual schema language, in accordance with ISO/TS 19103 and ISO 19109. These International Standards specify a framework for how to write application schemas. The rules include specifications on how to use standardized schemas to define feature types. It is necessary that both a sender and a receiver of data have access to the application schema.

The application schema shall be accessible to both ends of a data interchange to ensure a successful result. It is necessary that the application schema be transferred before data interchange takes place, so that both the receiver and sender can prepare their systems by implementing mappings and data structures according to the application schema. It may be transferred together with the dataset, or it may be stored in a public place and referenced from the dataset.

The application schema may be interchanged by paper- or electronic-based methods.

6.4 Encoding rule

6.4.1 Concept

An encoding rule is an identifiable collection of conversion rules that defines the encoding for a particular data structure. The encoding rule specifies the data types being converted, as well as the syntax, structure and coding schemes used in the resulting data structure. An encoding rule is applied to application-schema-specific data structures to produce system-independent data structures suitable for transport or storage. In order to define an encoding rule, it is necessary that three important aspects be specified: the input data structure, the output data structure and the conversion rules between the elements of the input and the output data structures. Both the input and output data structures are written using a conceptual schema language and the concepts in the languages are used to define the encoding rule.

6.4.2 Input data structure

The input data structure is an application-schema-specific data structure. The data structure can be thought of as a set of data instances, i.e. **i** $= \{i_1, ..., i_p\}$; see Figure 1. Each data instance, i_k, is an instance of a concept, I_l, defined in an application schema. The application schema defines a set of concepts defined in the application schema **I** $= \{I_1, ..., I_m\}$.

The application schema is a conceptual schema, **c**, written in a conceptual schema language, **C**. The conceptual schema defines a set of concepts **c** $= \{c_1, ..., c_m\}$ by instantiating the concepts of the conceptual schema language **C** $= \{C_1, ..., C_r\}$. Since the application schema is a conceptual schema, **c** = **I**.

6.4.3 Output data structure

The output data structure is defined by a schema, **D** $= \{D_1, ..., D_s\}$. D is the schema for the output structure and is not shown in Figure 1. The output data structure can be thought of as a set of data instances, i.e. **d** $= \{d_1, ..., d_q\}$ where each data instance, d_k, is an instance of a concept, D_l.

The schema, **D**, defines the syntax, structure and coding schemes of the output data structure.

6.4.4 Conversion rules

A conversion rule specifies how a data instance in the input data structure shall be converted to zero, one, or more instances in the output data structure. The conversion rules are defined and based on the concepts of the conceptual schema language, **C**, and on the concepts of the output data structure schema, **D**. It is necessary to specify a conversion rule, R_i, for each of the legal combinations of concepts in the conceptual schema language. The set of conversion rules are **R** $= \{R_1, ..., R_n\}$, where R_i is the i-th conversion rule and C_i is the i-th legal combination of instances from the schema language. A conversion table for all possible C_i can be set up, where each C_i maps to a production of instances in the output data structure, **D**. Figure 2 shows the relationship between the input and output conceptual schema language and the encoding rule.

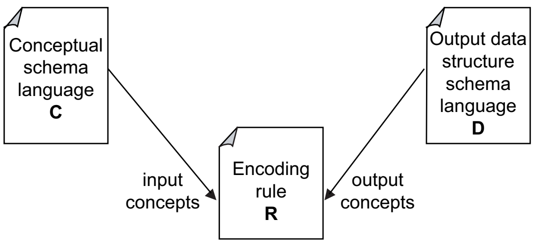

Figure 2 — The encoding rule defines conversion rules from input concepts to output concepts

NOTE The conversion rules are defined based on the two schema languages and not on any particular application schema. This is a generic approach that allows developers to write application-schema-independent encoding services, which can be used for different application schemas as long as the schemas are defined in the same conceptual schema language.