INTERNATIONAL STANDARD

ISO 19145

First edition
2013-02-15

Geographic information — Registry of representations of geographic point location

Information géographique — Registre de représentations de localisation de point géographique

Reference number
ISO 19145:2013(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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.

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 19145 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Introduction

ISO 6709:2008 standardizes the mechanisms for the interoperability of geographic point location representations. However, the representation of geographic point locations takes various schemes (e.g. ISO 6709:1983, DCMI Point encoding scheme, KML, GeoVRML, Natural Area Coding System, ISO 8211, GML Point Profile) depending of the application in which they are used. Accordingly, ISO 6709:2008 recognizes and supports flexibility in the representation of geographic point locations and the requirement for universal interpretation. In order to support the use of a variety of geographic point location representations, ISO 6709:2008 introduces the requirement of a registry of geographic point location representations. A registry of representations of geographic point location gives access to the description of the format in which a geographic point location is encoded and also identifies conversion services to transform the representation of the geographic point location to another representation. As such, knowing in which format a geographic point location is encoded and the format in which it must be encoded for its use by a specific application, it can be possible to perform the appropriate transformation of the representation of a geographic point location. However, this requires that encoding formats and their descriptions need to be made accessible either as part of the geographic point location representation itself or from a registry of representations of geographic point locations. As such, the definition of a standard structure for a registry of representations of geographic point location is required. Such a registry will support the required flexibility identified in ISO 6709:2008 for efficient syntactic interoperability of geographic point location information.

This International Standard defines a standard structure of a register in Unified Modelling Language (UML) that supports the description of geographic point location representation (Clause 7). It also defines the XML implementation of the register's UML structure by extending ISO/TS 19135-2, Annex A. Although the structure for the description of geographic point location representation takes its roots in ISO 19135, it extends that International Standard with specific requirements to an extent that it goes beyond the definition of a profile of ISO 19135.

Geographic information — Registry of representations of geographic point location

1 Scope

This International Standard specifies the process for establishing, maintaining and publishing registers of representation of geographic point location in compliance with ISO 19135. It identifies and describes the information elements and the structure of a register of representations of geographic point location including the elements for the conversion of one representation to another.

This International Standard also specifies the XML implementation of the required XML extension to ISO/TS 19135-2, for the implementation of a register of geographic point location representations.

A registry of geographic point location representations differs from a coordinate reference system (CRS) registry as it is not intended to describe the parameters of a CRS including datum, projections, units of measure, and order of coordinates but is concerned by the manner a geographic point location according to ISO 6709 is physically represented in a record or part of it.

2 Conformance

To conform to this International Standard, a register of geographic point location representations shall satisfy all of the conditions specified in the abstract test suite (Annex B).

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

ISO 19115:2003, *Geographic information — Metadata*

ISO 19118:2011, *Geographic information — Encoding*

ISO 19135:2005, *Geographic information — Procedures for item registration*

ISO/TS 19135-2:2012, *Geographic information - Procedures for item registration — Part 2: XML schema implementation*

ISO/TS 19139:2007, *Geographic information — Metadata — XML schema implementation*

W3C XMLName, *Namespaces in XML 1.0 (Second Edition).* W3C Recommendation (16 August 2006)

W3C XMLSchema-1, *XML Schema Part 1: Structures Second Edition. W3C Recommendation* (28 October 2004)

W3C XMLSchema-2, *XML Schema Part 2: Datatypes Second Edition. W3C Recommendation* (28 October 2004)

W3C XML, *Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation* (16 August 2006)

W3C XLink, *XML Linking Language (XLink) Version 1.0. W3C Recommendation* (27 June 2001)

4 Terms, definitions and abbreviations

4.1 Terms and definitions

4.1.1
compression
technique used for the reduction of space used by data

4.1.2
compression service
service (4.1.16) that accomplishes *compression* (4.1.1)

4.1.3
conversion
transformation from one *format* (4.1.9) to another

4.1.4
conversion service
service (4.1.16) that invokes a *converter* (4.1.5)

4.1.5
converter
resource that performs *conversion* (4.1.3)

Note 1 to entry: The resource can be a device or software.

4.1.6
coordinate
one of a sequence of *n* numbers designating the position of a point in *n*-dimensional space

Note 1 to entry: In a coordinate reference system, the coordinate numbers are qualified by units.

[SOURCE: ISO 19111:2007, 4.5]

4.1.7
coordinate tuple
tuple (4.1.18) composed of a sequence of *coordinates* (4.1.6)

Note 1 to entry: In a coordinate reference system, the coordinate numbers are qualified by units.

[SOURCE: ISO 19111:2007, 4.12, modified — Note 1 to entry has been added.]

4.1.8
dynamic conversion
online and real time *conversion* (4.1.3) of data

4.1.9
format
language construct that specifies the representation, in character form, of data objects in a record, file, message, storage device, or transmission channel

[SOURCE: ISO/IEC 2382-15:1999, 15.04.35]

4.1.10
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth

[SOURCE: ISO 19101:2002, 4.16]

4.1.11
geographic point location
well defined geographic place described by one *coordinate tuple* (4.1.7)

4.1.12
geographic point location representation
syntactic description of a *geographic point location* (4.1.11) in a well known *format* (4.1.9)

4.1.13
identifier
linguistically independent sequence of characters capable of uniquely and permanently identifying that with which it is associated

[SOURCE: ISO 19135:2005, 4.1.5]

4.1.14
register
set of files containing *identifiers* (4.1.13) assigned to items with descriptions of the associated items

[SOURCE: ISO 19135:2005, 4.1.9]

4.1.15
registry
information system on which a *register* (4.1.14) is maintained

[SOURCE: ISO 19135:2005, 4.1.13]

4.1.16
service
distinct part of the functionality that is provided by an entity through interfaces

[SOURCE: ISO 19119:2005, 4.1]

4.1.17
static conversion
offline process to perform a global *conversion* (4.1.3) of a large amount of data

4.1.18
tuple
ordered list of values

[SOURCE: ISO 19136:2007, 4.1.63]

4.2 Abbreviations

CRS	coordinate reference system
DCMI	Dublin Core Metadata Initiative
GeoVRML	Geo- Virtual Reality Modelling Language
GIS	geographic information system
GML	Geography Markup Language
GPL	geographic point location
GPLR	geographic point location representation
KML	Keyhole Markup Language
RFID	radio frequency identification
UML	Unified Modelling Language
XML	eXtensible Markup Language

5 Role of a register of representations of geographic point location

5.1 Overview

The exchange of geographic point locations (GPLs) described by coordinates might use various representations or formats. To use such information properly in applications, a GPL must be clear about the representation with which it complies. Then, conversion mechanisms can be applied to transform exchanged GPLs into systems' internal representations for their appropriate usage assuming that the internal representation is also registered and services are available.

Registers give the flexibility to manage geographic point location representations (GPLRs). Registers of GPLRs made publicly available as a file or web service enhance the interoperability of GPLs by clearly identifying how one GPL is represented and how it can be converted in another representation.

This clause highlights the role of registers of GPLRs for geographic information interoperability, especially for the conversion of one GPLR into another through different environments, including static vs. dynamic.

5.2 User's environment and registers

The role of a register invoked by a user's environment is depicted in Figure 1. In a user's environment, a *GIS* application typically gets its input data from an *external data* repository. Usually, that data needs to be converted into the internal representation of the user's GIS environment. This is made possible through a *geographic point location converter* service, which first searches in a register environment for possible transformation (i.e. *search for specifications*) and requests to the register environment the required information to transform the GPLR into the user's GIS environment (i.e. *request for specification*). To this end, the *geographic point location converter* service passes the identification of the GPLR from the external data source to the register's environment.

In the register's environment, it is the *register* service that receives the request. Using the identification of the GPLR passed by the conversion service, it gets the specification of the representation from the register (i.e. *database of geographic point location representation*) including the possible conversions into other representations and replies to the conversion service. Finally, the conversion service gets the description information (i.e. *geographic point location specifications*) of the representation including

known conversions to other representation and the *geographic point location converter* service will decide which one bests fit the user's GIS environment.

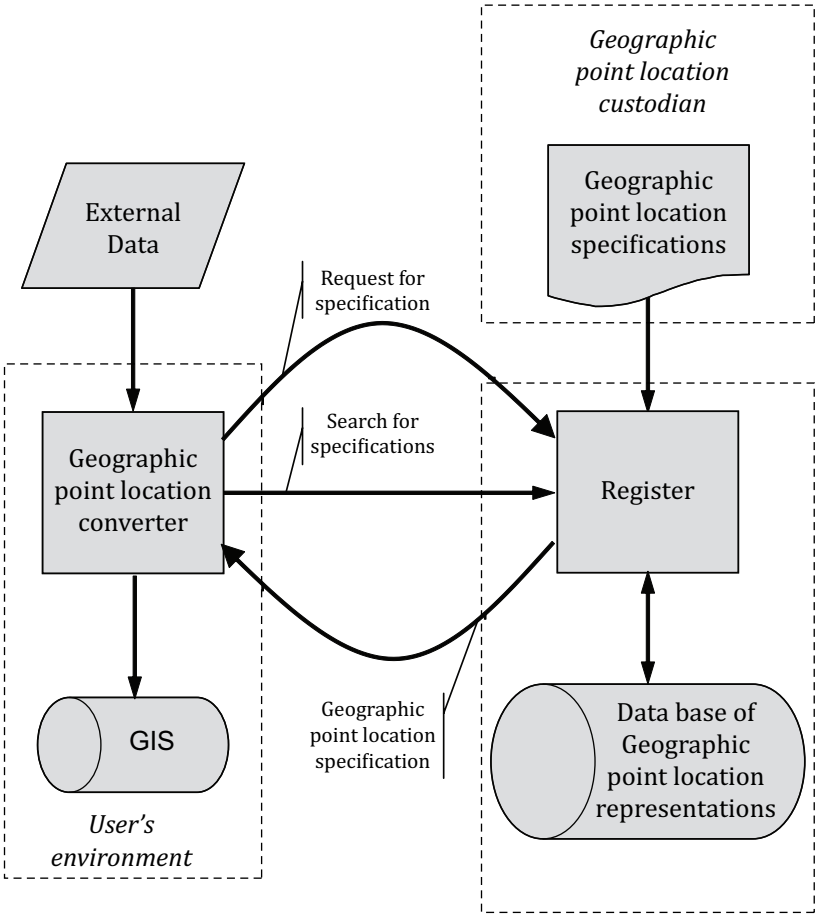

Figure 1 — User's environment and register

5.3 Static and dynamic data conversion

Two types of data conversion mechanism are recognized: static conversion and dynamic conversion.

Static data conversion consists usually in an offline process to perform a global conversion of a large amount of data. One such conversion is performed in batch conversion of static geographic data sets such as digital maps from one format to another. Another consists in the fusion process that integrates multiple geographic data sets represented by various formats in a common data set under a unique format.

Dynamic data conversion refers to an online and real-time processing mechanism. This means that a GPL can be imported or exported through a wireless network, converted, and used by a mobile terminal such as in telematics. For example, the location of moving features can be tracked globally by RFID sensors passing through the distributed RFID gateways where the various GPL may be managed in and converted from various coordinate systems and representations.

Consequently, a register of representations of geographic point locations serves in static and dynamic conversions as a resource to identify the representation format in which a GPL is encoded, to support the decoding of the GPL by the application (online or offline), and to support the encoding process in another format representation as required.

6 Management of registries and registers

Several actors are needed in the maintenance of registers, each one playing specific roles. ISO 19135 identifies and describes the following actors:

— Control body;

— Register manager;

— Register owner;

— Registry manager;

— Submitting organization.

The management of registries and registers shall be as specified in ISO 19135.

7 Schema of the register

7.1 Context

Clause 7 specifies the content and structure of the register of GPLR in a UML schema (see Annex C for a summary of UML notation). The schema includes only one package but refers to classes from the ISO/TS 19103 Conceptual Schema Language package, the ISO 19115 Metadata package, and the ISO 19135 Procedures for registration package (Figure 2). It includes information about the register itself and information about representations of geographic point locations (Figure 3).

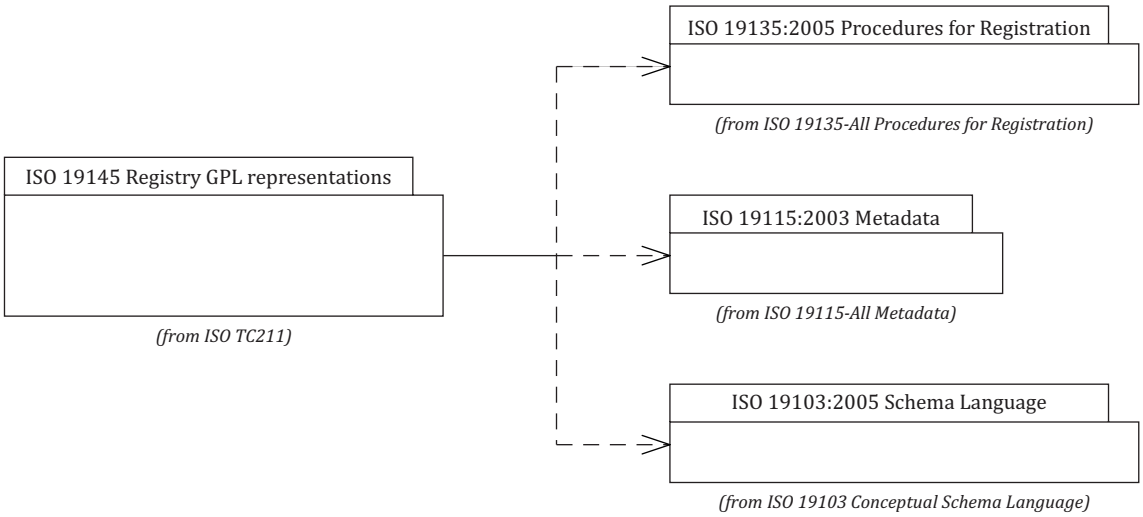

Figure 2 — GPLR package dependencies

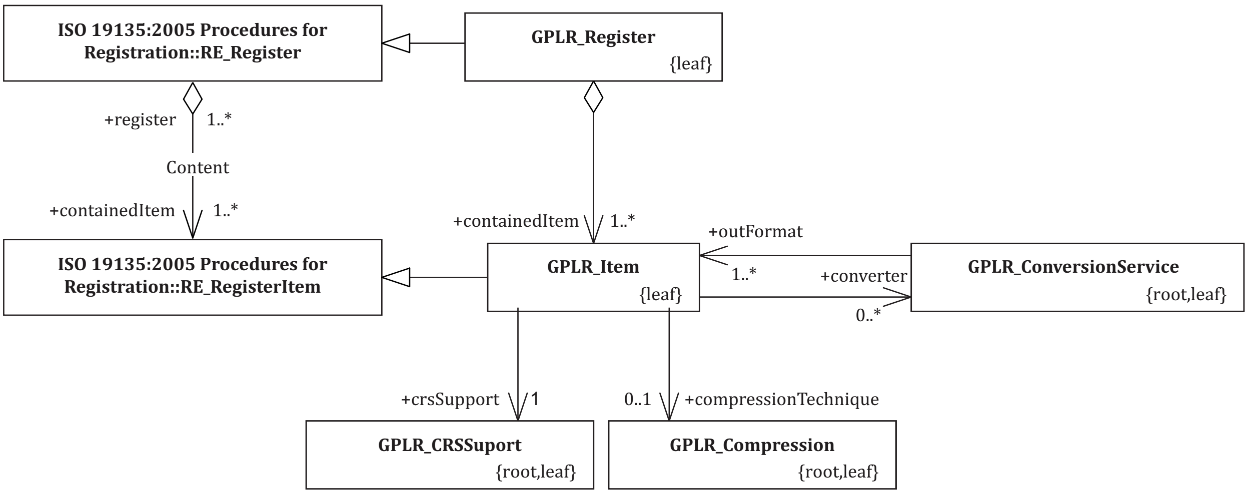

Figure 3 — Overview of the UML GPLR register's schema

7.2 GPLR_Register

7.2.1 Semantics

The class *GPLR_Register* (Figure 4) specifies information about a register of geographic point location representations. The GPLR_Register class is essentially a subclass of RE_Register from ISO 19135 and as such inherits all its properties, relations, and behaviours (see ISO 19135:2005, 8.2).

7.2.2 Content

The association *Content* aggregates GPLR_Items that are parts of the GPLR_Register. This association is essentially the counterpart of the Content association between RE_Register and RE_RegisterItem of ISO 19135.

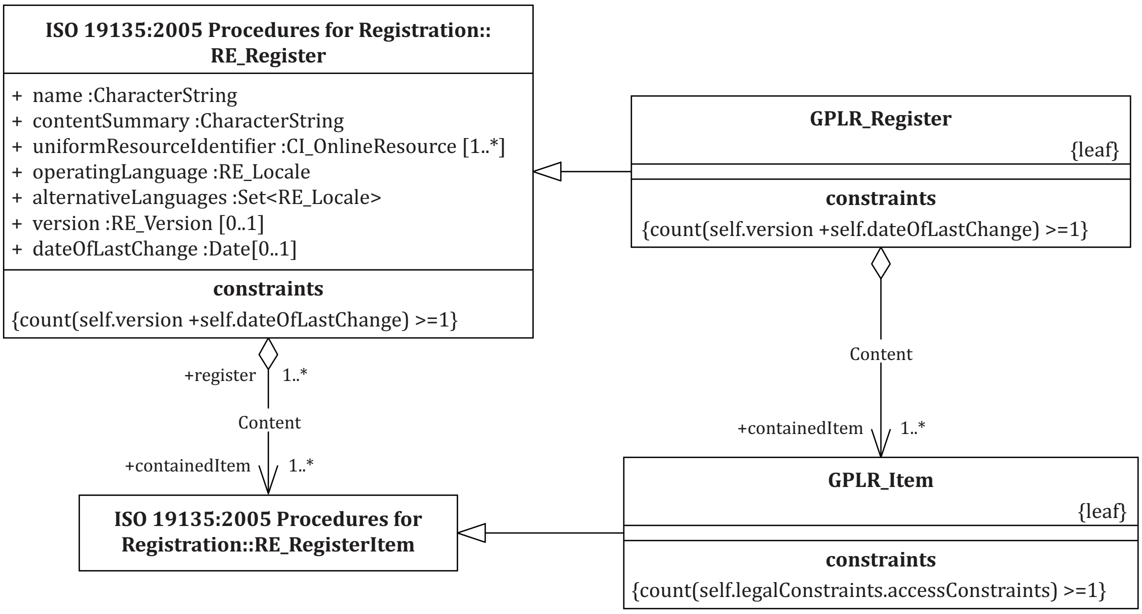

Figure 4 — GPLR_Register

7.3 GPLR_Item

7.3.1 Semantics

The class *GPLR_Item* (Figure 5) specifies information that describes geographic point location representations. The *GPLR_Item* class is essentially a subclass of RE_RegisterItem from ISO 19135 and as such inherits all its properties, relations, and behaviours (see ISO 19135:2005, 8.8). It adds five attributes and three association roles: *gplrID*, *specificationName*, *version*, *legalConstraints*, *type*, *crsSupport*, *converter*, and *compressionTechnique*. This information provides the details on how a geographic point location is rendered, compressed, and can be converted into another representation.

7.3.2 gplrID

The mandatory attribute *gplrID* consists in a designator that uniquely distinguishes from others a specific geographic point location representation within the context of the register. It is intended to be used for information processing. It is represented by a CharacterString (see ISO/TS 19103).

7.3.3 specificationName

The mandatory attribute *specificationName* designates a specific geographic point location representation. A specific geographic point location representation is represented by a CI_Citation (see ISO 19115). In CI_Citation, the attribute *title* identifies the specific name of the geographic point location representation and the attribute *citedResponsibleParty* provides the details of the resources related to the geographic point location representation. In the context of this International Standard, these two attributes are mandatory.

7.3.4 version

The mandatory attribute *version* provides information about the specific release of the geographic point location representation. It is represented by a GPLR_Version.

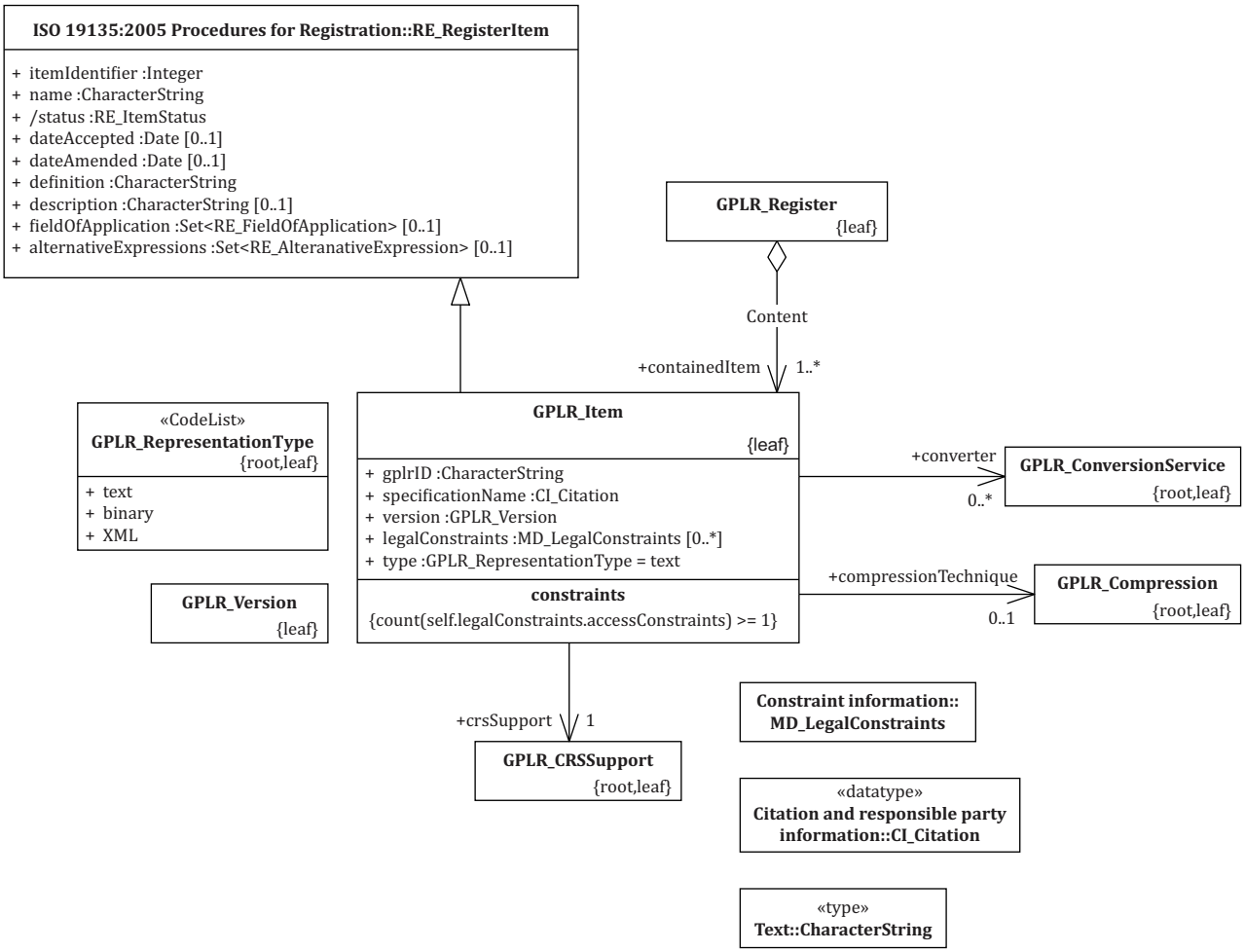

Figure 5 — GPLR_Item

7.3.5 legalConstraints

The optional attribute *legalConstraints* identifies whether or not the geographic point location representation can be used only under legal rights, restrictions, and prerequisites. It shall use an MD_LegalConstraints type as specified in ISO 19115. When a legal constraint is documented, at least *accessConstraints* must be documented.

7.3.6 type

The mandatory attribute *type* provides the encoding type of a geographic point location representation. It uses the code lists GPLR_RepresentationType. Accordingly, a GPL representation can be encoded as *text*, *binary*, or *XML*.

7.3.7 compressionTechnique

The optional association role *compressionTechnique* identifies the mechanism used to reduce the size of the geographic point location representation. It is described by a GPLR_Compression.

7.3.8 converter

The optional association role *converter* identifies the conversion tools or services that can transform this geographic point location representation into another one. It uses a GPLR_ConversionService.

7.3.9 crsSupport

The mandatory association role *crsSupport* identifies if the geographic point location representation provides support to coordinate reference systems. It is described by a GPLR_CRSSupport.

7.4 Geographic Point Location Representation Object

A geographic point location register is simple, i.e. it contains a single item class. Accordingly, the register has one unique *containedItem* of the type of RE_ItemClass, which is *Geographic Point Location Representation*. The GPLR_Register and all *containedItems* of the GPLR_Item class refer to this *Geographic Point Location Representation*. This is illustrated in Figure 6.

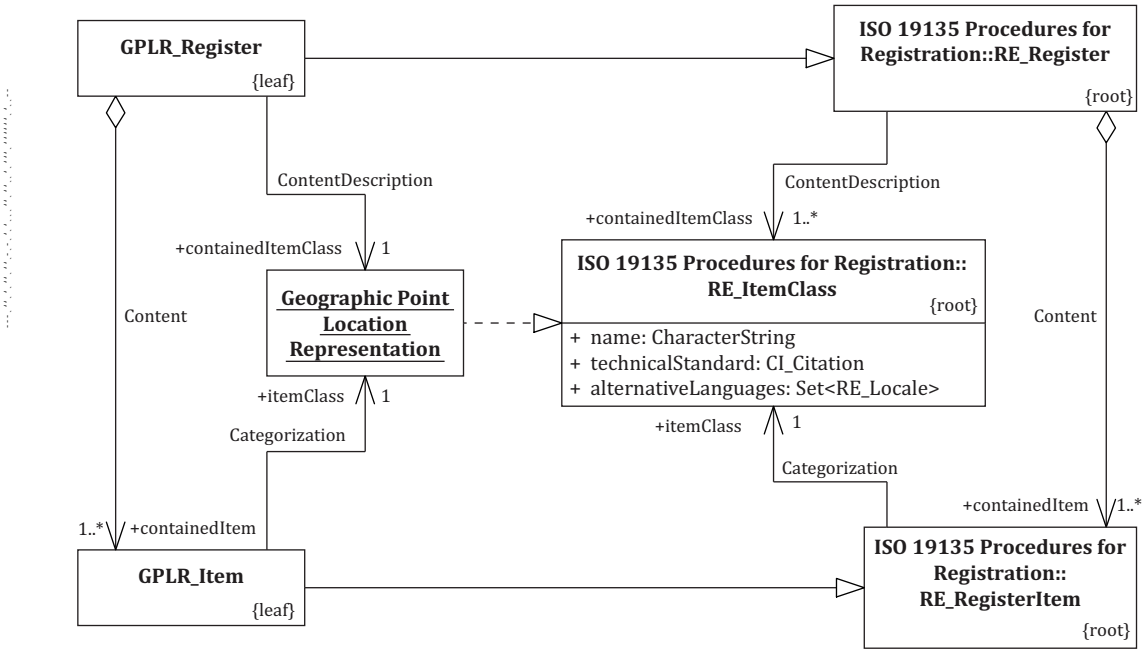

Figure 6 — Geographic Point Location Representation Object

7.5 GPLR_CRSSupport

7.5.1 Semantics

The class *GPLR_CRSSupport* (Figure 7) specifies information about the support of coordinate reference systems by the register using two attributes and one association role: *supported*, *fixed*, and *crs*.

7.5.2 supported

The mandatory derived attribute *supported* identifies whether or not the geographic point location representation supports at least one coordinate reference system. It shall use a Boolean type as specified in ISO/TS 19103. If a *crs* instance is associated with a GPLR_CRSSupport instance, then the attribute *supported* is set to *true*, otherwise it is set to *false*.

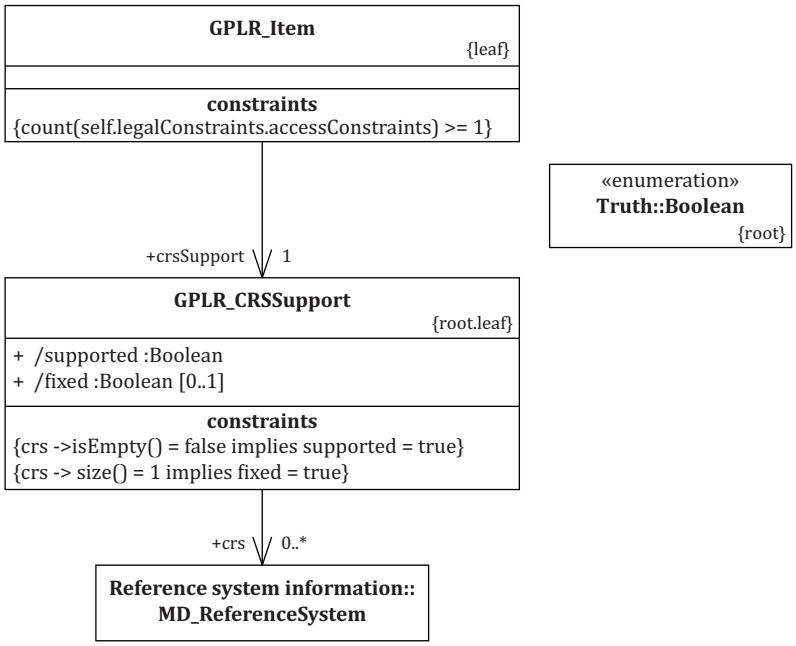

Figure 7 — GPLR_CRSSupport

7.5.3 fixed

If coordinate reference system is supported, the optional derived attribute *fixed* when set to *true* identifies that the geographic point location representation supports only one coordinate reference system, and when set to *false* identifies that the geographic point location representation supports multiple coordinate reference systems. It uses a Boolean type (see ISO/TS 19103).

7.5.4 crs

The optional association role *crs* identifies the coordinate reference systems that are supported by the geographic point location representation. It uses an MD_ReferenceSystem type (see ISO 19115).

7.6 GPLR_ConversionService

7.6.1 Semantics

The class *GPLR_ConversionService* (Figure 8) specifies information about converters to transform a geographic point location representation into another. The class uses four attributes and two association roles: *name*, *description*, *uniformResourceIdentifier*, *conversionType*, *invocationInterface* and *outFormat*.

7.6.2 name

The mandatory attribute *name* contains a human readable designator that denotes a conversion tool or a conversion service. A conversion service name shall be unique within a register. It is represented by a CharacterString (see ISO/TS 19103).

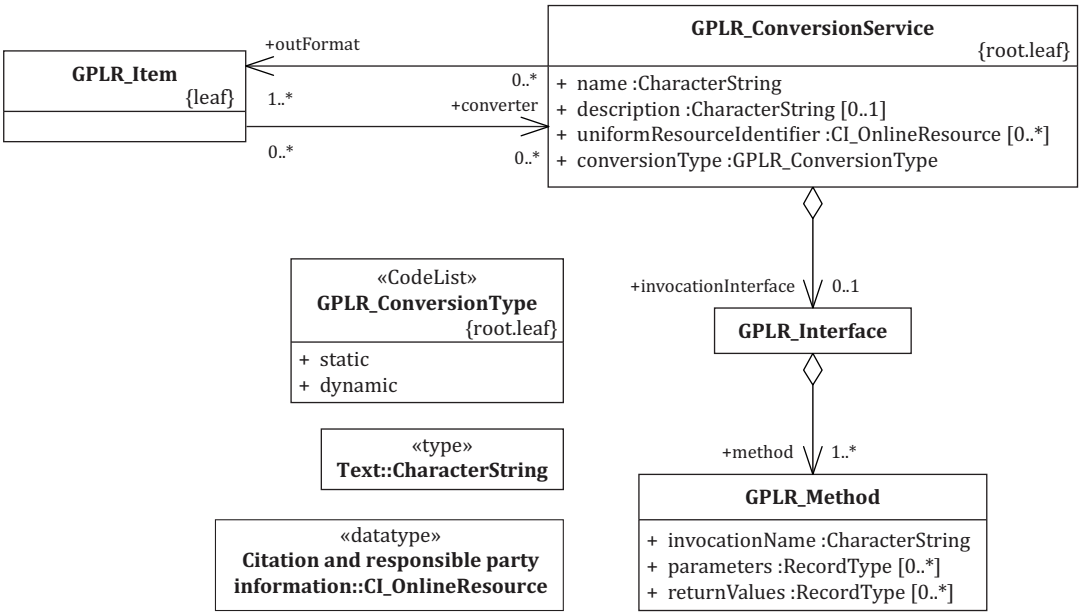

Figure 8 — GPLR_ConversionService

7.6.3 description

The optional attribute *description* provides a narrative overview or brief explanation of the conversion tool or conversion service. The description shall inform on the various processes that are involved in the conversion. It uses a CharacterString (see ISO/TS 19103).

7.6.4 uniformResourceIdentifier

The optional attribute *uniformResourceIdentifier* provides URLs to online references of resources associated with the conversion tool or conversion service. It is represented by a CI_OnlineResource (see ISO 19115).

7.6.5 conversionType

The mandatory attribute *conversionType* identifies whether the conversion is processed online or offline in batch mode for example. It uses the GPLR_ConversionType code list. Accordingly, a GPLR_ ConversionService can be identified *static* or *dynamic*.

7.6.6 invocationInterface

The optional association role *invocationInterface* describes the interaction mechanism by which the conversion service can be invocated. An invocation interface is documented by a GPLR_Interface.

7.6.7 outFormat

The mandatory association role *outFormat* identifies the various geographic point location representations that the conversion tool or conversion service can generate. Each outFormat instance is described by a GPLR_Item.

7.7 GPLR_Interface

7.7.1 Semantics

The class *GPLR_Interface* (Figure 8) specifies the methods that compose the invocation interface of the conversion.

7.7.2 method

The mandatory association role *method* identifies the various methods that are included in a GPLR_Interface. A method is described by a GPLR_Method.

7.8 GPLR_Method

7.8.1 Semantics

The class *GPLR_Method* (Figure 8) specifies the elements that describe a method of an interface. It has three attributes: *invocationName*, *parameters*, and *returnValues*.

7.8.2 invocationName

The mandatory attribute *invocationName* is a human readable designator that denotes the method. A method name shall be unique within an interface. It is represented by a CharacterString (see ISO/TS 19103).

7.8.3 parameters

The optional attribute *parameters* describe the input values involved in the method. Parameters are represented by a RecordType that specifies names and types of the parameters involved (see ISO/TS 19103).

7.8.4 returnValues

The optional attribute *returnValues* describes the output values returned by the method. returnValues are represented by a RecordType that specifies names and types of the elements returned (see ISO/TS 19103).

7.9 GPLR_Compression

7.9.1 Semantics

The class *GPLR_Compression* (Figure 9) specifies information about the compression technique that a specific geographic point location representation uses. It has four attributes: *name*, *description*, *version*, and *compressionService*.

7.9.2 name

The mandatory attribute *name* designates a specific compression mechanism. It is represented by a CI_Citation (see ISO 19115). In CI_Citation, the attribute *title* identifies the specific name of the compression mechanism and the attribute *citedResponsibleParty* provides details about resources related to the compression mechanism.

7.9.3 description

The optional attribute *description* provides a narrative overview or brief explanation of the compression technique. It uses a CharacterString (see ISO/TS 19103).

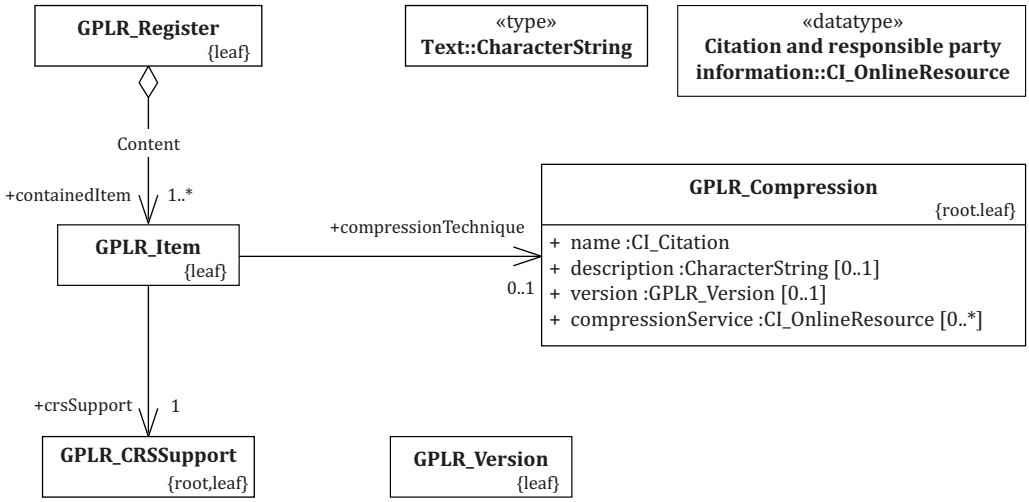

Figure 9 — GPLR_Compression

7.9.4 version

The optional attribute *version* provides information about the release of the compression technique. It is represented by a GPLR_Version.

7.9.5 compressionService

The optional attribute *compressionService* provides URLs to online services or downloadable tools that perform the compression. Each instance of compressionService is represented by a CI_OnlineResource (see ISO 19115).

7.10 GPLR_Version

7.10.1 Semantics

The class *GPLR_Version* (Figure 10) specifies information about the release of a resource. It inherits from RE_Version (see ISO 19135) and adds to the two RE_Version attributes one additional attribute to include a possible amendment to a given version. Therefore the class uses three attributes: *versionNumber*, *versionDate* and *amendmentNumber*.

7.10.2 versionNumber

The mandatory attribute *versionNumber* contains a constrained CharacterString (see ISO/TS 19103) that identifies the version and consists of a < first positive integer > < . > < second positive integer > < letters > :

a) < first positive integer > : major version designation

b) < . > : delimiter

c) < second positive integer > : optional minor version designation

d) < letters > : optional minor sub-version designation

EXAMPLE 2.1a

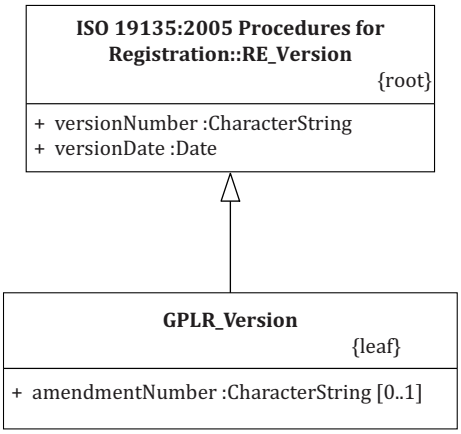

Figure 10 — GPLR_Version

7.10.3 versionDate

The mandatory attribute *versionDate* specifies the date of the release. It is represented by a Date (see ISO/TS 19103).

7.10.4 amendmentNumber

The optional attribute *amendmentNumber* contains a CharacterString (see ISO/TS 19103) that denotes a specific amendment associated with the resource version.

7.11 GPLR_RepresentationType

GPLR_RepresentationType is a code list that specifies the encoding method used for the representation of the geographic point location.

Table 1 — Encoding method used for the representation of the geographic point location

Value	Semantics
text	The geographic point location is represented in text format other than XML, but using a text encoding method such as in ISO/IEC 10646.
binary	The geographic point location is represented in binary format, i.e. not using a text encoding method.
XML	The geographic point location is represented in XML format.

7.12 GPLR_ConversionType

GPLR_ConversionType is a code list that specifies the processing mechanism used to convert geographic point location representations.

Table 2 — Processing mechanism used to convert geographic point location representations

Value	Semantics
static	Offline process to perform a global conversion of a large amount of data. [a]
dynamic	Online and real-time processing mechanism.
[a] It includes batch conversion of static geographic data sets from one format to another and integration of multiple geographic data sets from various formats in one common data set in a unique format.	

Annex A
(normative)

XML extensions required for registry of representations of geographic point location

A.1 XML schema implementation

A register of items of representation of geographic point location is described in UML as one specific package and has dependencies with other packages (see Clause 7). This annex provides an XML implementation for the encoding of ISO 19145 compliant registers, which extends on the XML implementation of ISO 19135, defined in ISO/TS 19135-2. The XML schema defined in this annex shall conform to the rules outlined in ISO 19118 and ISO/TS 19139.

The XML implementation of this International Standard defines the namespace http://www.isotc211.org/2005/gplr (Figure A.1) where the namespace prefix gplr stands for Geographic Point Location Representation.

The details of the XML description of all the parts that form an ISO 19145 XML register are provided in B.2.

A.2 XML Schema of the registry of representations of geographic point location

A.2.1 Geographic point location representation item

A.2.1.1 Introduction

A.2.1 documents the XML schema of the classes that are included in the description of the register of representation of geographic point location.

Figure A.1 shows the namespace used to implement ISO 19145 along with its dependencies on the namespaces defined in ISO/TS 19135-2 and ISO/TS 19139, used to implement ISO 19135 and ISO 19115 and the related standards, respectively.

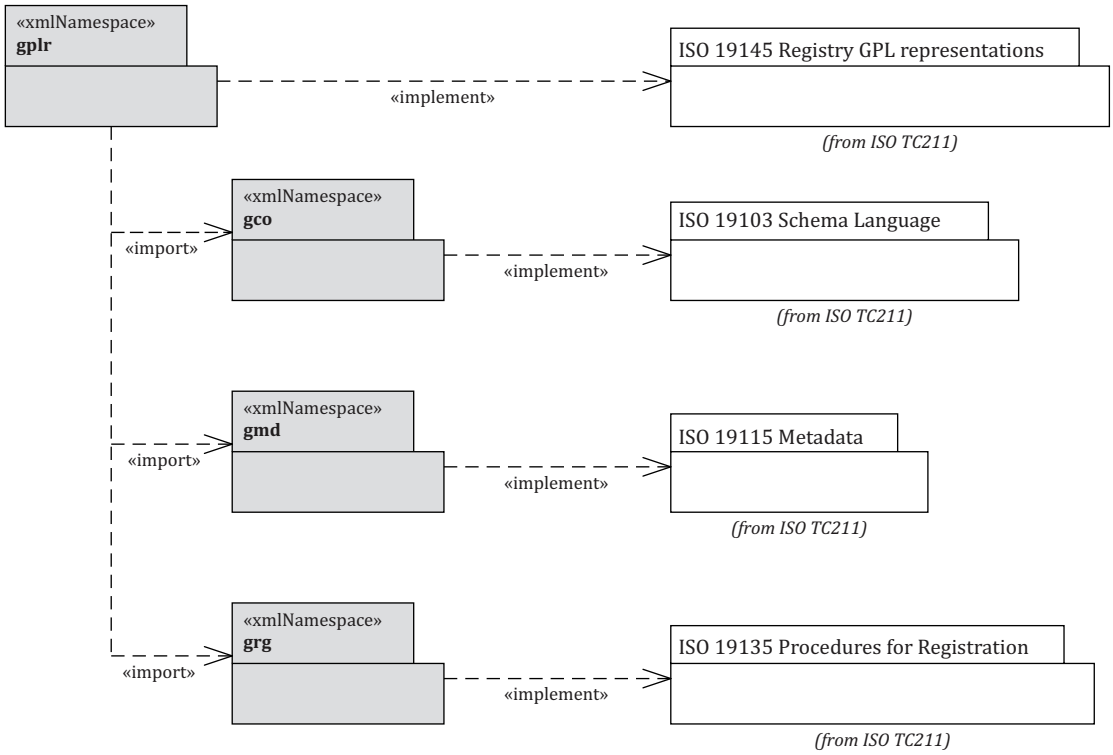

Figure A.1 — XML namespaces

For the XML schema element definition hereafter, the following illustrates the typical schema header:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.
w3.org/1999/xlink"
        xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gmd="http://www.isotc211.
org/2005/gmd"
        xmlns:grg="http://www.isotc211.org/2005/grg" xmlns:gplr="http://www.isotc211.
org/2005/gplr"
        targetNamespace="http://www.isotc211.org/2005/gplr" elementFormDefault="qualified"
        version="0.1">
<!- ================================== Annotation ================================ →
 <xs:annotation>
  <xs:documentation>This file was generated from ISO 19145 UML class diagrams</
xs:documentation>
 </xs:annotation>
<!- ============================== Imports & Includes ============================== →
 <xs:import namespace="http://www.isotc211.org/2005/gco"
        schemaLocation="http://standards.iso.org/ittf/
PubliclyAvailableStandards/ISO_19139_Schemas/
        gco/gco.xsd"/>
 <xs:import namespace="http://www.isotc211.org/2005/gmd"
        schemaLocation="http://standards.iso.org/ittf/
PubliclyAvailableStandards/ISO_19139_Schemas/
        gmd/gmd.xsd"/>
 <xs:import namespace="http://www.isotc211.org/2005/grg"
        schemaLocation="http://standards.iso.org/ittf/
PubliclyAvailableStandards/ISO_19135-
        2_Schemas/grg.xsd"/>
</xs:schema>
```

A.2.1.2 Namespace organization

The namespace gplr contains the implementation of the schema of the register of representations of geographic point location. The root of this namespace is gplr.xsd. Figure A.2 shows the organization of the gplr namespace. gplrRegister shall include the element definitions described in A.2.1.3 to A.2.1.11 and A.2.2.

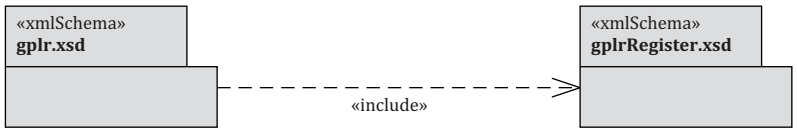

Figure A.2 — gplr namespace

A.2.1.3 GPLR_Item

The following XML schema defines the GPLR_Item_Type complex type, the GPLR_Item element and the GPLR_Item_PropertyType complex type. These definitions correspond to the GPLR_Item class of the UML model.

```
<xs:complexType name="GPLR_Item_Type">
  <xs:annotation>
   <xs:documentation>Element of information recorded in the register</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="grg:RE_RegisterItem_Type">
    <xs:sequence>
     <xs:element name="gplrID" type = "gco:CharacterString_PropertyType"/>
     <xs:element name="specificationName" type = "gmd:CI_Citation_PropertyType"/>
     <xs:element name="version" type = "gplr:GPLR_Version_PropertyType"/>
     <xs:element name="legalConstraints" type = "gmd:MD_LegalConstraints_PropertyType"
minOccurs="0"
          maxOccurs="unbounded"/>
     <xs:element name="type" type="gplr:GPLR_RepresentationType_PropertyType"/>
     <xs:element name="crsSupport" type="gplr:GPLR_CRSSupport_PropertyType"/>
     <xs:element name="converter" type="gplr:GPLR_ConversionService_PropertyType" minOc-
curs="0"
          maxOccurs="unbounded"/>
     <xs:element name="compressionTechnique" type="gplr:GPLR_Compression_PropertyType"
          minOccurs="0"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_Item" type = "gplr:GPLR_Item_Type"
          substitutionGroup="grg:RE_RegisterItem"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_Item_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_Item"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.4 GPLR_CRSSupport

The following XML schema defines the GPLR_CRSSupport_Type complex type, the GPLR_CRSSupport element, and the GPLR_CRSSupport_PropertyType complex type. These definitions correspond to the GPLR_CRSSupport class of the UML model.

```
<xs:complexType name="GPLR_CRSSupport_Type">
  <xs:annotation>
```

```
   <xs:documentation>Information about the support of a coordinate reference system</
xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="gco:AbstractObject_Type">
    <xs:sequence>
     <xs:element name="supported" type="gco:Boolean_PropertyType"/>
     <xs:element name="fixed" type="gco:Boolean_PropertyType" minOccurs="0"/>
     <xs:element name="crs" type="gmd:MD_ReferenceSystem_PropertyType" minOccurs="0"
           maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_CRSSupport" type="gplr:GPLR_CRSSupport_Type"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_CRSSupport_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_CRSSupport"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.5 GPLR_Method

The following XML schema defines the GPLR_Method_Type complex type, the GPLR_Method element, and the GPLR_Method_PropertyType complex type. These definitions correspond to the GPLR_Method class of the UML model.

```
<xs:complexType name="GPLR_Method_Type">
 <xs:annotation>
  <xs:documentation>Elements that describe a method of an interface</xs:documentation>
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="gco:AbstractObject_Type">
   <xs:sequence>
    <xs:element name="invocationName" type="gco:CharacterString_PropertyType"/>
    <xs:element name="parameters" type="gco:RecordType_PropertyType" minOccurs="0"
           maxOccurs="unbounded"/>
    <xs:element name="returnValues" type="gco:RecordType_PropertyType" minOccurs="0"
           maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<!- ..................................................................... →
<xs:element name="GPLR_Method" type="gplr:GPLR_Method_Type"/>
<!- ..................................................................... →
<xs:complexType name="GPLR_Method_PropertyType">
 <xs:sequence minOccurs="0">
  <xs:element ref="gplr:GPLR_Method"/>
 </xs:sequence>
 <xs:attributeGroup ref="gco:ObjectReference"/>
 <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

A.2.1.6 GPLR_Interface

The following XML schema defines the GPLR_Interface_Type complex type, the GPLR_Interface element, and the GPLR_Interface_PropertyType complex type. These definitions correspond to the GPLR_Interface class of the UML model.

```
<xs:complexType name="GPLR_Interface_Type">
 <xs:annotation>
```

```
  <xs:documentation>Methods that compose the invocation interface</xs:documentation>
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="gco:AbstractObject_Type">
   <xs:sequence>
    <xs:element name="method" type="gplr:GPLR_Method_PropertyType" maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<!- ..................................................................... →
<xs:element name="GPLR_Interface" type="gplr:GPLR_Interface_Type"/>
<!- ..................................................................... →
<xs:complexType name="GPLR_Interface_PropertyType">
 <xs:sequence minOccurs="0">
  <xs:element ref="gplr:GPLR_Interface"/>
 </xs:sequence>
 <xs:attributeGroup ref="gco:ObjectReference"/>
 <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

A.2.1.7 GPLR_ConversionService

The following XML schema defines the GPLR_ConversionService_Type complex type, the GPLR_ ConversionService element, and the GPLR_ConversionService_PropertyType complex type. These definitions correspond to the GPLR_ConversionService class of the UML model.

```
<xs:complexType name="GPLR_ConversionService_Type">
  <xs:annotation>
   <xs:documentation>Information about a conversion service</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="gco:AbstractObject_Type">
    <xs:sequence>
     <xs:element name="name" type="gco:CharacterString_PropertyType"/>
     <xs:element name="description" type="gco:CharacterString_PropertyType" minOc-
curs="0"/>
     <xs:element name="uniformResourceIdentifier" type="gmd:CI_OnlineResource_PropertyType"
          minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="conversionType" type="gplr:GPLR_ConversionType_PropertyType"/>
     <xs:element name="invocationInterface" type="gplr:GPLR_Interface_PropertyType" minOc-
curs="0"/>
     <xs:element name="outFormat" type="gplr:GPLR_Item_PropertyType" maxOc-
curs = "unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_ConversionService" type="gplr:GPLR_ConversionService_Type"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_ConversionService_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_ConversionService"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.8 GPLR_Compression

The following XML schema defines the GPLR_Compression_Type complex type, the GPLR_Compression element, and the GPLR_Compression_PropertyType complex type. These definitions correspond to the GPLR_Compression class of the UML model.

```
<xs:complexType name="GPLR_Compression_Type">
  <xs:annotation>
   <xs:documentation>Information about a compression technique</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="gco:AbstractObject_Type">
    <xs:sequence>
     <xs:element name="name" type="gmd:CI_Citation_PropertyType"/>
     <xs:element name="description" type="gco:CharacterString_PropertyType" minOc-
curs="0"/>
     <xs:element name="version" type="gplr:GPLR_Version_PropertyType" minOccurs="0"/>
     <xs:element name="compressionService" type="gmd:CI_OnlineResource_PropertyType"
minOccurs="0"
          maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_Compression" type = "gplr:GPLR_Compression_Type"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_Compression_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_Compression"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.9 GPLR_Version

The following XML schema defines the GPLR_Version_Type complex type based on the RE_Version_Type of the ISO 19135:2005 XML schema, the GPLR_Version element, and the GPLR_Version_PropertyType complex type. These definitions correspond to the GPLR_Version class of the UML model.

```
<xs:complexType name="GPLR_Version_Type">
  <xs:annotation>
   <xs:documentation>Information about a release of a resource</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="grg:RE_Version_Type">
    <xs:sequence>
     <xs:element name="amendmentNumber" type="gco:CharacterString_PropertyType" minOc-
curs="0"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_Version" type="gplr:GPLR_Version_Type"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_Version_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_Version"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.10 GPLR_RepresentationType

The following XML schema defines the GPLR_RepresentationType element and the GPLR_RepresentationType_PropertyType complex type that correspond to the GPLR_RepresentationType codelist of the UML model.

```
<xs:element name="GPLR_RepresentationType" type="gco:CodeListValue_Type"
        substitutionGroup="gco:CharacterString"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_RepresentationType_PropertyType">
  <xs:sequence>
   <xs:element ref="gplr:GPLR_RepresentationType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.1.11 GPLR_ConversionType

The following XML schema defines the GPLR_ConversionType element and the GPLR_ConversionType_ PropertyType complex type that correspond to the GPLR_ConversionType codelist of the UML model.

```
<xs:element name="GPLR_ConversionType" type="gco:CodeListValue_Type"
        substitutionGroup="gco:CharacterString"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_ConversionType_PropertyType">
  <xs:sequence>
   <xs:element ref="gplr:GPLR_ConversionType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

A.2.2 Geographic point location representation register

A.2.2 documents the XML schema of the geographic point location representation register class. It defines the GPLR_Register_Type complex type, the GPLR_Register element, and the GPLR_Register_ PropertyType complex type that correspond to the GPLR_Register class of the UML model.

```
<xs:complexType name="GPLR_Register_Type">
  <xs:annotation>
   <xs:documentation>Information about and XML structure of the geographic point location
           register</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="grg:RE_Register_Type">
    <xs:sequence>
     <xs:element name="containedItem" type="gplr:GPLR_Item_PropertyType"
maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
<!- ..................................................................... →
 <xs:element name="GPLR_Register" type="gplr:GPLR_Register_Type"
          substitutionGroup="grg:RE_Register"/>
<!- ..................................................................... →
 <xs:complexType name="GPLR_Register_PropertyType">
  <xs:sequence minOccurs="0">
   <xs:element ref="gplr:GPLR_Register"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

Annex B
(normative)

Abstract test suite

B.1 Introduction

This normative annex presents the abstract test suite for evaluating conformance to this International Standard. The abstract test suite contains the following test modules: management procedures (B.2), register content (B.3), publication of register contents (B.4), publication of registers in XML (B.5) and by-Value, by reference, gco:nilReason (B.6).

B.2 Management procedures

a) Test Purpose: Verify that the register of geographic point location representations is managed according to the rules specified in this International Standard.

b) Test Method: Check the procedures described in the information distributed by the registration manager.

c) Reference: Clause 6 and ISO 19135:2005, Clause 6.

d) Test Type: Capability.

B.3 Register content

a) Test Purpose: Verify that the register of geographic point location representations contains the minimum specified content.

b) Test Method: Inspect entries in the register to ensure that they include all elements of information required by this International Standard.

c) Reference: Clause 7.

d) Test Type: Capability.

B.4 Publication of register contents

a) Test Purpose: Verify that the contents of the register of geographic point location representations are publicly available.

b) Test Method: Check the information distributed by the registry manager. Visit the website and inspect the information made available.

c) Reference: Clause 6 and ISO 19135:2005, 6.4.

d) Test Type: Capability.

B.5 Publication of registers in XML

a) Test Purpose: Verify that the contents of the XML instance document of the register of geographic point location representations validate without error against the XML schemas.

b) Test Method: Validate the XML instance document against the XML schemas using a tool that implements a strict interpretation of the W3C XML schema recommendation.

c) Reference: Annex A.

d) Test Type: Capability.

B.6 By-Value, by-reference, gco:nilReason

a) Test Purpose: Verify that the contents of XML elements is either expressed by value (i.e. explicit content described between a start-tag and an end-tag), by reference (i.e. using an attribute that points to the resource), or by a NULL reason.

b) Test Method: Validate the XML instance document using mechanisms such as XSL transformations and Schematron, or by human inspection.

c) Reference: Annex A.

d) Test Type: Capability.

Annex C
(informative)

UML notation

C.1 Introduction

This annex provides a brief description of UML notation as specified in ISO/IEC 19501 and ISO/TS 19103, and as used in the UML diagrams in this International Standard.

C.2 Class

A UML class (Figure C.1) represents a concept within the system being modelled. It is a description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A class is drawn as a solid-outline rectangle with three compartments separated by horizontal lines. The top name compartment holds the class name and other general properties of the class (including stereotype); the middle list compartment holds a list of attributes; the bottom list compartment holds a list of operations. The attribute and operation compartments may be suppressed to simplify a diagram. Suppression does not indicate that there are no attributes or operations.

NOTE This International Standard does not specify any operations or methods.

ISO/TS 19103 specifies that a class name shall include no blank spaces and that individual words in the name shall begin with capital letters.

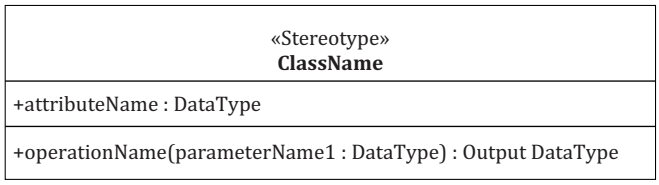

Figure C.1 — UML class

C.3 Stereotype

Stereotypes extend the semantics, but not the structure of pre-existing types and classes. A stereotype is used to classify (or mark) other UML elements so that they behave as if they were instances of new "virtual" metamodel classes whose form is based on existing "base" classes. A stereotype may introduce additional values and additional constraints. All model elements that are classified by a particular stereotype receive these values and constraints.

Class level stereotypes used in this International Standard include:

1) <<Type>> specified in ISO/IEC 19501, specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects. However, it may have attributes and associations.

2) <<DataType>> specified in ISO/IEC 19501, is a descriptor of a set of values that lack identity (independent existence and the possibility of side effects). Data types include primitive predefined types and user-definable types. A DataType is thus a class with few or no operations

whose primary purpose is to hold the abstract state of another class for transmittal, storage, encoding or persistent storage.

3) <<Enumeration>> specified in ISO/IEC 19501, is a data type whose instances form a list of named literal values. Both the enumeration name and its literal values are declared. Enumeration means a short list of well-understood potential values within a class. Classic examples are Boolean that has only two (or three) potential values TRUE, FALSE (and NULL). Most enumerations will be encoded as a sequential set of Integers, unless specified otherwise. The actual encoding is normally only of use to programming language compilers.

4) <<CodeList>> , specified in ISO/TS 19103, is a flexible enumeration that uses string values through a binding of the Dictionary type key and returns values as string types, e.g. Dictionary (String, String). A CodeList is useful for expressing a long list of potential values. If the elements of the list are completely known, an Enumeration shall be used; if only the likely values of the elements are known, a codeList shall be used. Enumerated code lists may be encoded according to a standard, such as ISO 3166-1. CodeLists are more likely to have their values exposed to the user, and are therefore often mnemonic. Different implementations are likely to use different encoding schemes (with translation tables to other encoding schemes available).

C.4 Attribute

An attribute represents a characteristic common to the objects of a class. An attribute is specified by a text string that can be parsed into elements that describe the properties of the attribute:

visibility name [multiplicity]: type-expression = initial-value

where:

visibility may be public (indicated by "+") or private (indicated by "–").

name is a character string. ISO/TS 19103 specifies that an attribute name shall include no blank spaces, that it shall begin with a lower case letter, and that individual words in the name, following the first word, shall begin with upper case letters.

multiplicity specifies the number of values that an instance of a class may have for a given attribute. The notation is explained in C.10. When multiplicity of an attribute is not shown in a diagram, it has the default value of 1.

type-expression identifies the data type of the attribute.

initial-value, if present, specifies a default value for the attribute.

C.5 Association

An association (Figure C.2) is a semantic relationship between classes that specifies connections between their instances. An association is drawn as a solid line connecting two class rectangles. An association may have a name, represented as a character string placed near the line, but not close to either end. ISO/TS 19103 specifies that an association name shall include no blank spaces and that individual words in the name shall begin with upper case letters. The association ends are adorned with information pertinent to the class at that end of the association, including multiplicity and role name.

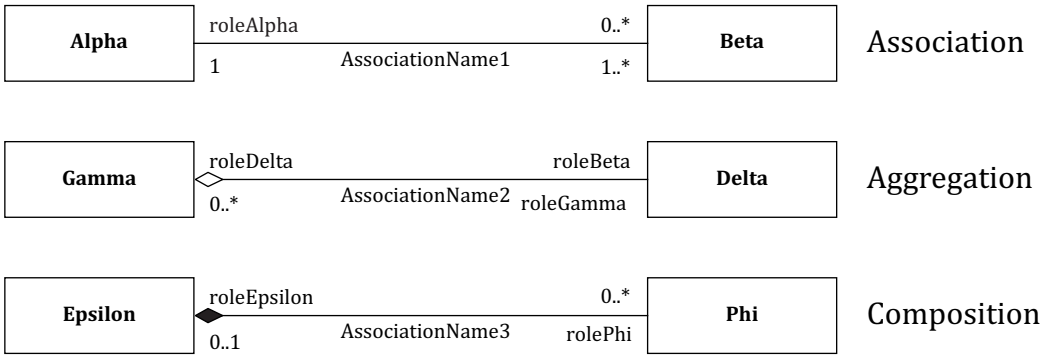

Figure C.2 — UML associations

C.6 Role name

A role name adorning an association end specifies behaviour of the class at that end with respect to the class at the other end of the association. In Figure C.2, roleAlpha describes the role that the class named Alpha has with respect to the class named Beta. A role name is represented as a character string. ISO/TS 19103 specifies that a role name shall include no blank spaces, that it shall begin with a lower case letter, and that individual words in the name, following the first word, shall begin with upper case letters.

C.7 Navigability

An arrow attached to the end of an association path indicates that navigation is supported toward the class attached to the arrow. In other words, information held in that class is accessible from the class at the other end of the association. Arrows may be attached to zero, one, or two ends of the path. This International Standard follows the practice of showing arrows only in the case of association paths that are navigable in only one direction. All other associations are assumed to be navigable in both directions. In Figure C.2, AssociationName2 is navigable from Gamma to Delta, but not in the opposite direction.

C.8 Aggregation

Associations may be used to show aggregation or composition relationships between classes. An open diamond on an association end indicates that the class at that end of the association is an aggregate of instances of the class at the other end of the association. For example, the class named Gamma, in Figure C.2, is an aggregate of zero or more instances of the class named Delta. Aggregation is considered a weak form of composition. The members of an aggregation can exist independently of the aggregation, and can be members of more than one aggregation.

C.9 Composition

A closed diamond on an association end indicates that the class at that end of the association is composed of instances of the class at the other end of the association. For example, the class named Epsilon in Figure C.2 is composed of zero or more instances of the class named Phi. Members of a composite cannot exist independently of the composite class, nor can they be members of more than one composite class.

C.10 Multiplicity

Multiplicity specifies the number of instances of a class that may be associated with a class at the other end of the association.

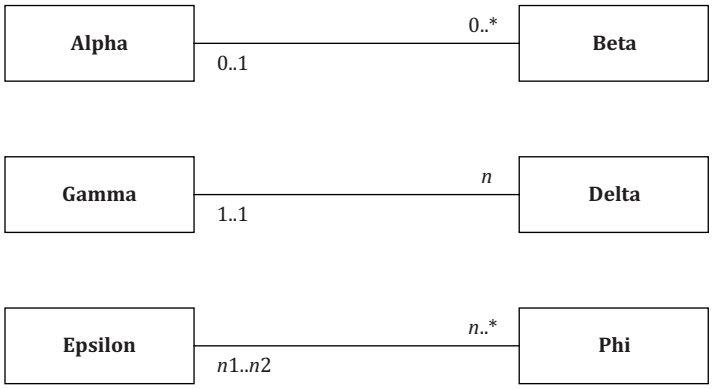

Figure C.3 — UML multiplicity

The values shown in Figure C.3 are all valid. They have the following meanings:

— zero or one instance of Alpha may be associated with one instance of Beta;

— zero or more instances of Beta may be associated with one instance of Alpha;

— one and only one instance of Gamma may be associated with one instance of Delta;

— *n* being an integer number, *n* and only *n* instances of Delta may be associated with one instance of Gamma;

— *n1* and *n2* being integer numbers, with *n2* > *n1*, the number of instances of Epsilon that may be associated with an instance of Phi may be within the range *n1* to *n2*;

— *n* being an integer number, *n* or more instances of Phi may be associated with one instance of Epsilon.

C.11 Generalization

ISO/IEC 19501 defines generalization (Figure C.4) as a taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed. Generalization is shown as a solid-line path from the child (the more specific element, such as a subclass) to the parent (the more general element, such as a superclass), with a large hollow triangle at the end of the path where it meets the more general element. Figure C.4 shows two generalization relationships.

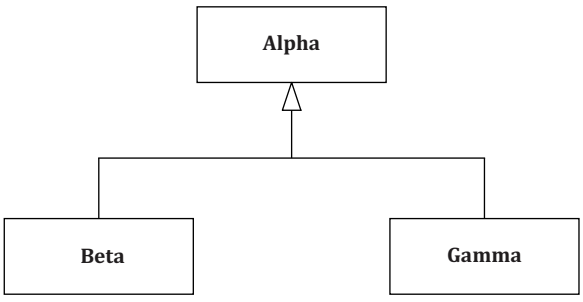

Figure C.4 — UML generalization

C.12 Derived elements

A derived element, such as an attribute or a rolename, is one whose value can be computed from another element, but is shown for clarity even though it adds no semantic information. A derived element is indicated by a slash ("/") in front of its name.

C.13 Note

A note (Figure C.5) contains textual information. It is shown as a rectangle with a "bent corner" in the upper right corner, attached to zero or more model elements by a dashed line. Notes may be used to contain comments or constraints.

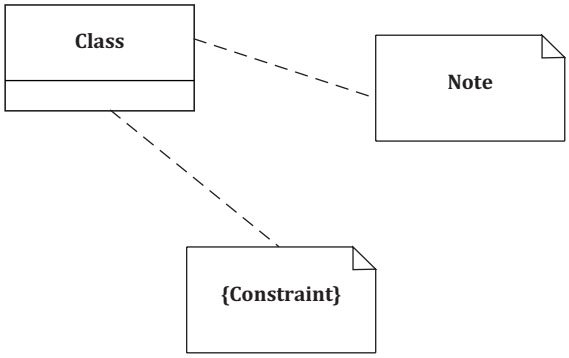

Figure C.5 — Note and constraint

C.14 Constraint

A constraint specifies a semantic condition or restriction. Although ISO/IEC 19501 specifies an Object Constraint Language for writing constraints, a constraint may be written using any formal notation, or a natural language. A constraint is shown as a text string in braces ("{ }"). It is placed near the element to which it applies. If the notation for an element is a text string (such as an attribute), the constraint string may follow the element text string in braces. A constraint included as an element in a list applies to all subsequent elements in the list, down to the next constraint element or the end of the list.

Bibliography

[1] ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

[2] ISO 6709:2008, *Standard representation of geographic point location by coordinates*

[3] ISO/IEC 8211, *Information technology — Specification for a data descriptive file for information interchange*

[4] ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

[5] ISO 19101, *Geographic information — Reference model*

[6] ISO 19111, *Geographic information — Spatial referencing by coordinates*

[7] ISO 19119, *Geographic information — Services*

[8] ISO 19127, *Geographic information — Geodetic codes and parameters*

[9] ISO 19136, *Geographic information — Geography Markup Language (GML)*

[10] ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[11] ISO/IEC 2382-15, *Information technology — Vocabulary — Part 15: Programming languages*

[12] ISO/TC 211 *Geographic information — Review summary of the preliminary work item on Project 19145 Geographic information — Registry of representations of geographic point location* (N 2357)

[13] Open Geospatial Consortium, Inc., *GML Point Profile* (OGC 05-029r4)

ICS 35.240.70

Price based on 30 pages