DRAFT FOR DEVELOPMENT

DD CEN ISO/TS 19139:2009

Geographic information — Metadata — XML schema implementation

ICS 35.240.70

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI Documents
Document Sharing Hub
British Standards

Full document details: DD CEN ISO/TS 19139:2009

National foreword

This Draft for Development is the UK implementation of CEN ISO/TS 19139:2009. It is identical to ISO/TS 19139:2007. It supersedes DD ISO/TS 19139:2007 which is withdrawn.

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature. It should be applied on this provisional basis, so that information and experience of its practical application can be obtained.

Comments arising from the use of this Draft for Development are requested so that UK experience can be reported to the European organization responsible for its conversion to a European standard. A review of this publication will be initiated two years after its publication by the European organization so that a decision can be taken on its status. Notification of the start of the review period will be made in an announcement in the appropriate issue of *Update Standards*.

According to the replies received by the end of the review period, the responsible BSI Committee will decide whether to support the conversion into a European Standard, to extend the life of the Technical Specification or to withdraw it. Comments should be sent to the Secretary of the responsible BSI Technical Committee at British Standards House, 389 Chiswick High Road, London W4 4AL.

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

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

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

This Draft for Development was published under the authority of the Standards Policy and Strategy Committee on 29 June 2007

© BSI 2010

ISBN 978 0 580 64446 7

Amendments/corrigenda issued since publication

Date	Comments
28 February 2010	This corrigendum renumbers DD ISO/TS 19139:2007 as DD CEN ISO/TS 19139:2009

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

TECHNICAL SPECIFICATION

SPÉCIFICATION TECHNIQUE

TECHNISCHE SPEZIFIKATION

CEN ISO/TS 19139

November 2009

ICS 35.240.70

English Version

Geographic information - Metadata - XML schema implementation (ISO/TS 19139:2007)

Information géographique - Métadonnées - Implémentation de schémas XML (ISO/TS 19139:2007)

Geoinformation - Metadaten - XML-Schema-Implementierung (ISO/TS 19139:2007)

This Technical Specification (CEN/TS) was approved by CEN on 14 May 2009 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

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

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

Management Centre: Avenue Marnix 17, B-1000 Brussels

© 2009 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. CEN ISO/TS 19139 :2009: E

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

Foreword

The text of ISO/TS 19139:2007 has been prepared by Technical Committee ISO/TC 211 “Geographic information/Geomatics” of the International Organization for Standardization (ISO) and has been taken over as CEN ISO/TS 19139:2009 by Technical Committee CEN/TC 287 “Geographic Information” the secretariat of which is held by NEN.

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

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

Endorsement notice

The text of ISO/TS 19139:2007 has been approved by CEN as a CEN ISO/TS 19139:2009 without any modification.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

Contents

Page

Introduction iv
1 Scope 1
2 Conformance 1
3 Normative references 1
4 Terms and definitions 2
5 Symbols and abbreviated terms 3
5.1 Acronyms 3
5.2 Namespace abbreviations 3
5.3 UML model relationships 3
5.4 UML model stereotypes 4
6 Requirements 5
6.1 Introduction to gmd 5
6.2 Rule-based 5
6.3 Quality 6
6.4 Web implementations 6
6.5 Use of external XML implementations 6
6.6 Multilingual support 6
6.7 Polymorphism 7
6.8 Rules for application schema 7
7 Extensions to the UML models in the ISO 19100 series of International Standards 8
7.1 Introduction to extensions 8
7.2 Extensions specific to the web environment 8
7.3 Cultural and linguistic adaptability extensions 9
7.4 Extensions for metadata-based transfers of geospatial information 11
8 Encoding rules 17
8.1 Introduction to encoding rules 17
8.2 Default XML Class Type encoding 17
8.3 XML Class Global Element encoding 20
8.4 XML Class Property Type encoding 20
8.5 Special case encodings 22
8.6 XML namespace package encoding 40
8.7 XML schema package encoding 41
9 Encoding descriptions 43
9.1 Introduction to the encoding descriptions 43
9.2 XML namespaces 43
9.3 gmd namespace 44
9.4 gss namespace 50
9.5 gts namespace 52
9.6 gsr namespace 53
9.7 gco namespace 54
9.8 gmx namespace 65
9.9 From the conceptual schema to XML file instances 72
Annex A (normative) Abstract test suite 74
Annex B (normative) Data dictionary for extensions 77
Annex C (informative) Geographic Metadata XML resources 87
Annex D (informative) Implementation examples 89
Bibliography 111

AVA Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

Introduction

The importance of metadata describing digital geographic data is explained in detail in the text of ISO 19115. ISO 19115 provides a structure for describing digital geographic data by defining metadata elements and establishing a common set of metadata terminology, definitions and extension procedures. ISO 19115 is abstract in that it provides a worldwide view of metadata relative to geographic information, but no encoding.

Since ISO 19115 does not provide any encoding, the actual implementation of geographic information metadata could vary based on the interpretation of metadata producers. In an attempt to facilitate the standardization of implementations, this comprehensive metadata implementation specification provides a definitive, rule-based encoding for applying ISO 19115. This Technical Specification provides Extensible Markup Language (XML) schemas that are meant to enhance interoperability by providing a common specification for describing, validating and exchanging metadata about geographic datasets, dataset series, individual geographic features, feature attributes, feature types, feature properties, etc.

ISO 19115 defines general-purpose metadata in the field of geographic information. More detailed metadata for geographic data types and geographic services are defined in other ISO 19100 series standards and user extensions (ISO 19115). This Technical Specification is also intended to define implementation guidelines for general-purpose metadata. Where necessary, interpretations of some other ISO 19100 series standards are incorporated.

ISO 19118 describes the requirements for creating encoding rules based on UML schemas and the XML-based encoding rules as well as providing an introduction to XML. This Technical Specification utilizes the encoding rules defined in ISO 19118 and provides the specific details of their application with regard to deriving XML schema for the UML models in ISO 19115.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

Geographic information — Metadata — XML schema implementation

1 Scope

This Technical Specification defines Geographic MetaData XML (gmd) encoding, an XML schema implementation derived from ISO 19115.

2 Conformance

Conformance with this Technical Specification shall be checked using all the relevant tests specified in Annex A. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105.

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 639-2, *Codes for the representation of names of languages — Part 2: Alpha-3 code*

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*

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

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*

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

ISO 19105, *Geographic information — Conformance and testing*

ISO 19107, *Geographic information — Spatial schema*

ISO 19108, *Geographic information — Temporal schema*

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

ISO 19110, *Geographic information — Methodology for feature cataloguing*

ISO 19111:—[1)], *Geographic information — Spatial referencing by coordinates*

ISO 19115:2003, *Geographic information — Metadata*

---

1) To be published. (Revision of ISO 19111:2003)

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

ISO 19115:2003/Cor. 1:2006, *Geographic information — Metadata — Technical Corrigendum 1*

ISO 19117, *Geographic information — Portrayal*

ISO 19118:2005, *Geographic information — Encoding*

ISO 19136:—[2)], *Geographic information — Geography Markup Language (GML)*

W3C XMLName, *Namespaces in XML. W3C Recommendation* (14 January 1999)

W3C XMLSchema-1, *XML Schema Part 1: Structures. W3C Recommendation* (2 May 2001)

W3C XMLSchema-2, *XML Schema Part 2: Datatypes. W3C Recommendation* (2 May 2001)

W3C XML, *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation* (6 October 2000)

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

4 Terms and definitions

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

4.1
namespace
collection of names, identified by a URI reference, that are used in XML documents as element names and attribute names

[W3C XML]

4.2
package
general purpose mechanism for organizing elements into groups

[ISO/TS 19103, definition 4.2.22]

EXAMPLE Identification information; Metadata entity set information; Constraint information.

4.3
realization
semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out

[Booch 1999]

4.4
polymorphism
characteristic of being able to assign a different meaning or usage to something in different contexts – specifically, to allow an entity such as a variable, a function, or an object to have more than one form

NOTE There are several different kinds of polymorphism.

[http://searchsmallbizit.techtarget.com]

2) To be published.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

5 Symbols and abbreviated terms

5.1 Acronyms

UML Unified Modelling Language

XCT XML Class Type

XCPT XML Class Property Type

XCGE XML Class Global Element

XML Extensible Markup Language

XPath XML Path Language

XSD XML Schema Definition

XSL Extensible Style Language

XSLT XSL Transformation

5.2 Namespace abbreviations

In the lists below, the item on the left describes the common namespace prefix used to describe the elements in the namespace. The second item is an English description of the namespace prefix, and the item in parenthesis is the URI of the actual namespace. These URIs do not correspond necessarily to an effective location of the schemas.

This first list corresponds to the namespaces defined by this Technical Specification.

gco	Geographic Common extensible markup language	(http://www.isotc211.org/2005/gco)
gmd	Geographic MetaData extensible markup language	(http://www.isotc211.org/2005/gmd)
gmx	Geographic Metadata XML Schema	(http://www.isotc211.org/2005/gmx)
gss	Geographic Spatial Schema extensible markup language	(http://www.isotc211.org/2005/gss)
gsr	Geographic Spatial Referencing extensible markup language	(http://www.isotc211.org/2005/gsr)
gts	Geographic Temporal Schema extensible markup language	(http://www.isotc211.org/2005/gts)

This second list corresponds to external namespaces used by this Technical Specification.

gml	Geography Markup Language	(use the GML namespace URI stated in ISO 19136)
xlink	XML Linking Language	(use the XLINK namespace URI stated in the W3C XLink recommendation)
xs	W3C XML base schemas	(use the XML schema namespace URI stated in the W3C XMLSchema-1 and W3C XMLSchema-2 recommendations)

5.3 UML model relationships

The diagrams that appear in this Technical Specification are presented using the Unified Modelling Language (UML) as the conceptual schema language as defined in ISO/TS 19103. ISO 19115:2003, Figure 2, also displays the UML notation that is used to describe the metadata. In addition to the UML described in ISO/TS 19103 and shown in ISO 19115, this Technical Specification uses the notation shown in Figure 1.

AVA Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

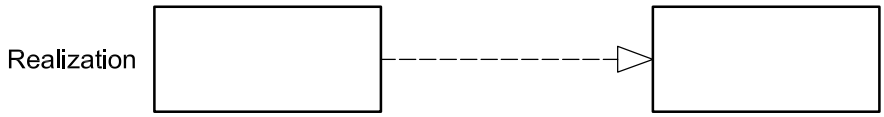

Figure 1 — UML Notation for Realization

The class that is the source of the connection (shown on the left in Figure 1) is guaranteed to carry out (or implement) the specification of the class at the destination of the connection (shown on the right in Figure 1).

5.4 UML model stereotypes

A UML stereotype is an extension mechanism for existing UML concepts (see ISO 19115). In addition to the stereotypes already defined in ISO 19115 for describing metadata, this Technical Specification defines stereotypes necessary for a rules-based encoding into XML schema.

The elements of the UML diagrams depicted in Clause 9 can carry stereotypes specifying an XML implementation. Those stereotypes are carried by classes representing XML elements or XML types, UML attributes, UML links (realizations or dependencies) and UML packages.

In this Technical Specification the following stereotypes of classes are used.

a) <<xs:choice>>: the class represents an implementation type encoded as an XML choice block. Each property of the class is implemented as an element of the choice.

b) <<xs:complexType>>: the class represents an implementation type encoded as an XML complex type.

c) <<xs:element>>: the class represents an XML global element.

d) <<xs:simpleType>>: the class represents an implementation type encoded as an XML simple type.

e) <<xs:simpleContent>>: the class represents an implementation type encoded as an XML complex type with simple content.

In this Technical Specification the following stereotypes of attributes are used.

f) <<xs:attribute>>: the property is encoded as an XML attribute.

g) <<xs:element>>: the property is encoded as an XML element with a name and a type (<xs:element name="propertyName" type="propertyType"/>)

h) <<pattern>>: this stereotype only applies to XML simple types derived from built-in XML types. A property with the stereotype <<pattern>> restricts the range of a built-in simple type.

i) <<XCGE>>: the property is encoded as a reference to an XML global element (<xs:element ref="XCGE"/>).

In this Technical Specification the following stereotypes of links are used.

j) <<XCT>>: (carried by realization relationships). The XCT of the abstract concept to implement is substituted by the specified external implementation.

k) <<XCGE>>: (carried by realization relationships). The XCGE of the abstract concept to implement is substituted by the specified external implementation.

l) <<XCPT>>: (carried by realization relationships). The XCPT of the abstract concept to implement is substituted by the specified external implementation.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

m) <<implement>>: (carried by dependency relationships). The source represents an XML schema implementing the abstract concepts defined in the target.

n) <<include>>: (carried by dependency relationships). The source and the target represent XML schemas. The source includes (<xs:include … />) the target.

o) <<import>>: (carried by dependency relationships). The source and the target represent sets of XML objects grouped within the same namespace. The source imports (<xs:import … />) the target.

In this Technical Specification the following stereotypes of packages are used.

p) <<xmlSchema>>: The package represents an XML schema.

q) <<xmlNamespace>>: The package represents a set of XML objects grouped within the same namespace.

6 Requirements

6.1 Introduction to gmd

Geographic metadata is represented in ISO 19115 as a set of UML packages containing one or more UML classes. ISO 19115 provides a universal, encoding-independent view of geographic information metadata. This Technical Specification provides a universal implementation of ISO 19115 through an XML schema encoding that conforms to the rules described in ISO 19118.

While the details of XML namespaces are not included in this Technical Specification, the contents of several namespaces are defined here. A namespace is really a collection of names which can be used in XML documents as element or attribute names. The namespace is used to identify the names with a particular schema. A namespace is a URI, and the ones utilized in this Technical Specification are listed in 5.2. A URI is often cumbersome for reading, writing and including in human discussion, so this Technical Specification will more often refer to common namespace prefixes when identifying particulars about the contents of a namespace. The primary namespace defined in this Technical Specification is http://www.isotc211.org/2005/gmd and the namespace prefix is **gmd** which stands for ***G**eographic **M**eta**D**ata extensible markup language*.

XML schema offers many alternatives for structuring information for exchange. ISO 19118 defines a set of encoding rules for transforming a UML conceptual schema from the ISO 19100 series of documents into an XML schema. Even within the pared down limitations of ISO 19118, there are still choices for the creation of specific XML schemas. Clauses 7, 8 and 9 describe the details of encoding the ISO 19115 UML conceptual schema and the UML models depicted in the respective ISO 19100 series of International Standards listed in Clause 3, into a set of XML schemas. A description of geographic ISO/TS 19139 XML resources and examples of metadata instance documents are included in Annexes C and D.

Before delving into the details of the encoding it is important to understand why certain encoding rules are utilized in the development of gmd. Gaining an understanding of the rules will make the capabilities, limitations and best-practice use of gmd clear. Some of the major goals for gmd were interoperability with other ISO 19100 series specifications, predictability, extensibility and usability. Further details of these goals are described in 6.2 to 6.8.

6.2 Rule-based

This XML schema implementation is a rule-based encoding built from the UML models in the ISO 19100 series of International Standards as required by ISO 19118. Using this methodology achieves a couple of the goals mentioned in 6.1. First, the resulting gmd schemas are based directly on other ISO 19100 series International Standards and therefore increase the chance for interoperability. Second, the resulting schema is predictable since any class, attribute, association, etc. is encoded just as other UML elements of the same type are encoded.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

Although not discussed in detail in this Technical Specification, having a rule-based encoding also allows the XML schemas to be generated in an automated or semi-automated fashion.

6.3 Quality

Quality in terms of the XML schema implementation implies simple XML schemas as well as the human readability of the XML files. The structural complexity of the ISO 19100 series, particularly ISO 19115, implies that it is not possible to provide simple XML schemas for geographic metadata. But the encoding rules are defined so that a user can directly create and/or understand the content of an XML instance document using the UML models in ISO 19115 as a basis. Additionally, an implementer can determine the XML schema implementation of the UML models in ISO 19115 by knowing the encoding rules.

Another aspect to the quality of gmd is completeness. This Technical Specification encodes the entire UML model from ISO 19115 without regard to a particular usage for gmd or a particular application schema that will utilize gmd.

6.4 Web implementations

One of the goals stated in 6.1 is usability. Usability, as it pertains to the design of gmd, focuses on the exchange of geographic metadata with the understanding that this will often happen in a web-like environment. While there is no restraint against creating geographic metadata instance documents based on gmd which never transfer across a network, there are many aspects to the design that are intended to aid internet and web-like transfer of data.

6.5 Use of external XML implementations

Another design principle that aids interoperability and usability is the use of existing XML schemas. If an XML schema standard already exists that encodes a part of the ISO 19100 series pertaining to geographic metadata then it is advantageous to incorporate that XML schema standard into gmd. If gmd uses the external XML schema directly, then interoperability is enhanced. It is also likely that software already exists that can process instance documents that conform to the external XML. Furthermore, if the external schema is well designed it might be more efficient than XML schema generated from a series of encoding rules and this might help achieve the goal of usability.

While using an implementation that already exists has some important advantages, the external XML schemas should not violate the primary design principles of gmd. For example, if an external XML schema implements part of the ISO 19100 series but does so in a cumbersome, unusable manner then it is not incorporated into gmd. Additionally, if an external XML schema does not readily meet the requirements stated in 6.6 for multilingual support then it is not incorporated into gmd.

6.6 Multilingual support

Cultural and linguistic adaptability is a basic requirement for any textual metadata elements. In Annex J of ISO 19115:2003, there is an informative discussion of multilingual textual metadata elements. In order to enhance the chances for interoperability of implementations it is important that a normative mechanism for multilingual support be included in this Technical Specification. The details of multilingual support are described in 7.3 and 9.8.6 but for the sake of understanding the design of gmd it is important to understand that special consideration is given to this requirement. The specific mechanisms used to achieve this goal are polymorphism and codelist registers. Polymorphism is introduced in 6.7 and codelist registers are described in 7.4.4.4.

It is also important to understand that the multilingual support that exists in W3C XML is not sufficient for the expression of geographic metadata. In W3C XML, “A special attribute named xml:lang may be inserted in documents to specify the language used in the contents and attribute values of any element in an XML document [W3C XML]”. If a particular element can only occur once based on the encoding rules discussed in Clause 8 then the technique of using the special xml:lang attribute to indicate the language does not allow for the specification of the same element in two or more languages.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

6.7 Polymorphism

The term polymorphism is formally defined in Clause 4. In general terms, polymorphism means the ability to assume different forms. In terms of this Technical Specification, the first obvious use for polymorphism is for providing cultural and linguistic adaptability. This allows implementers to provide geographic metadata in one or more languages without violating any cardinal rules defined in ISO 19115. Polymorphism provides more than just support for multiple languages. It also allows user communities to better refine geographic metadata to meet their organizational needs. For example, ISO 19115 contains an *individualName* attribute of type *CharacterString* in the *CI_ResponsibleParty* class, but within an organization individuals may be described in a more compartmentalized form (e.g. by first, middle and last names). Polymorphism allows implementers to extend the more general format of *individualName* within their namespaces while still fully utilizing gmd and still providing usable and understandable instance documents for users outside of their organization. The characteristics of gmd that allow for polymorphism primarily derive from the property type encodings described in 8.4.

6.8 Rules for application schema

ISO 19109 defines the rules for application schema and is comprised of two categories of models that are related to metadata:

— **a General Feature Model** that determines the particular way metadata and quality elements relate to geographic features;

— **two interchange models**: the traditional data transfer model and the interoperability model, each of them implying an interrelation of the metadata with its resources.

With regards to the ISO 19109 General Feature Model, the following specific types of attribute can be defined for a feature type:

— metadata attributes (as instances of GF_MetadataAttributeType) whose data type is MD_Metadata or one of its subclasses;

— quality attributes (as instances of GF_QualitiyAttributeType) whose data type is DQ_Element or one of its subclasses.

The use of geographic metadata XML schema in the context of these rules for application schema basically consists of the use of the XML schema definitions corresponding to the metadata and quality attribute data types. The use of XML schema definitions when encoding feature types using ISO 19136 is described in 9.9.2.

With regards to the ISO 19109 interchange models, the *interoperability* model is based on data interchange by transactions and is designed for a large number of transactions involving simple interchange. In contrast, the *transfer model* is designed for a lesser number of transactions with large amounts of well-organized data. The ISO 19115 metadata conceptual schema and this Technical Specification are clearly designed to be a starting point for providing more coherent transfer of geospatial data among and within user communities (i.e. *transfer model*).

Additionally, it is important to understand that the *interoperability* model applies to the interaction between the user application and a service provider, and that the interaction is fully determined by the service interface. While the service interface may not *require* the use of a specific schema, the adoption of the geographic metadata XML schema within an information-sharing community is highly recommended when applicable. Interoperable interchanges by transfer needs to go further in terms of standardization. This is the purpose of the “Extensions for metadata-based transfers of geospatial information” presented in 7.4.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

7 Extensions to the UML models in the ISO 19100 series of International Standards

7.1 Introduction to extensions

It has already been stated that ISO 19115 provides an internationally-accepted, encoding-independent view of geographic information metadata and that this Technical Specification provides a worldwide implementation of ISO 19115 through an XML schema encoding. Before delving into the specifics of the encoding it is important to recognise that once a specific implementation technology is identified there may be some specific extensions required for the ISO 19100 series UML models. The purpose of these extensions may vary but they are mainly intended to support the requirements stated in Clause 6. An extension might be created to facilitate interoperability, ease of use, web-like environments, etc. The UML diagrams shown throughout Clause 7 are the extensions identified to support an XML schema encoding of ISO 19115 and its related ISO 19100 series of International Standards.

7.2 Extensions specific to the web environment

There are several extensions of the CharacterString class from ISO/TS 19103 that are necessary to add convenience when working with XML documents. These extensions are specific to the World Wide Web environment where XML documents are typically processed. Figure 2 defines the metadata required to describe elements specific to working in a web environment. The data dictionary for this diagram is located in B.2.1.

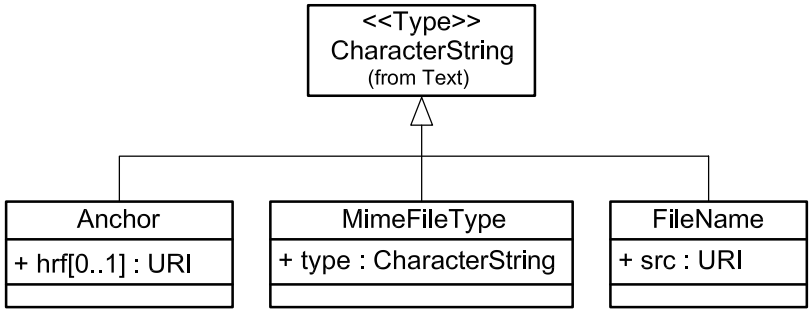

Figure 2 — Extensions to support web environments

The Anchor class is needed to support hyper-linking capabilities and to ensure a web-like implementation of ISO 19115's RS_Identifier and MD_Identifier classes. Because Anchor is a subclass of CharacterString, the *code* attribute of RS_Identifier and MD_Identifier can be implemented by instantiating Anchor and thus provide a reference to the place where the information related to the code is available.

The FileName class is needed to support explicitly referencing of an external file corresponding to a property containing the name of the file. This is valuable in the case of ISO 19115's MD_BrowseGraphic class with regards to the fileName attribute. A human readable file name might be a useful value for population of this attribute, but the additional src attribute provided by the FileName class can provide a machine-readable absolute path to the location of the file.

The MimeFileType class is needed to support identification of the file type using the mime media type name and subtype name. This is useful in the case of ISO 19115's MD_BrowseGraphic class with regards to the fileType attribute. The value of the fileType attribute might be “JPEG” and the type attribute of the MimeFileType class allows for the machine-readable Mime-type content-type expression such as “image/jpeg”.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

7.3 Cultural and linguistic adaptability extensions

7.3.1 Free text

The free text element in the domain of a CharacterString property type in ISO 19115 is intended to support a textual metadata element having multiple instances of the same information in different locales. A locale (identified as PT_Locale) is a combination of language, potentially a country, and a character encoding (i.e. character set) in which localized character strings are expressed. Annex J of ISO 19115:2003, describes this Free Text concept (identified as PT_FreeText) but does not include the conceptual schema to accompany the description. This Technical Specification makes the use of PT_FreeText normatively by providing a conceptual schema in Figure 3 as well as a corresponding data dictionary (see B.2.2).

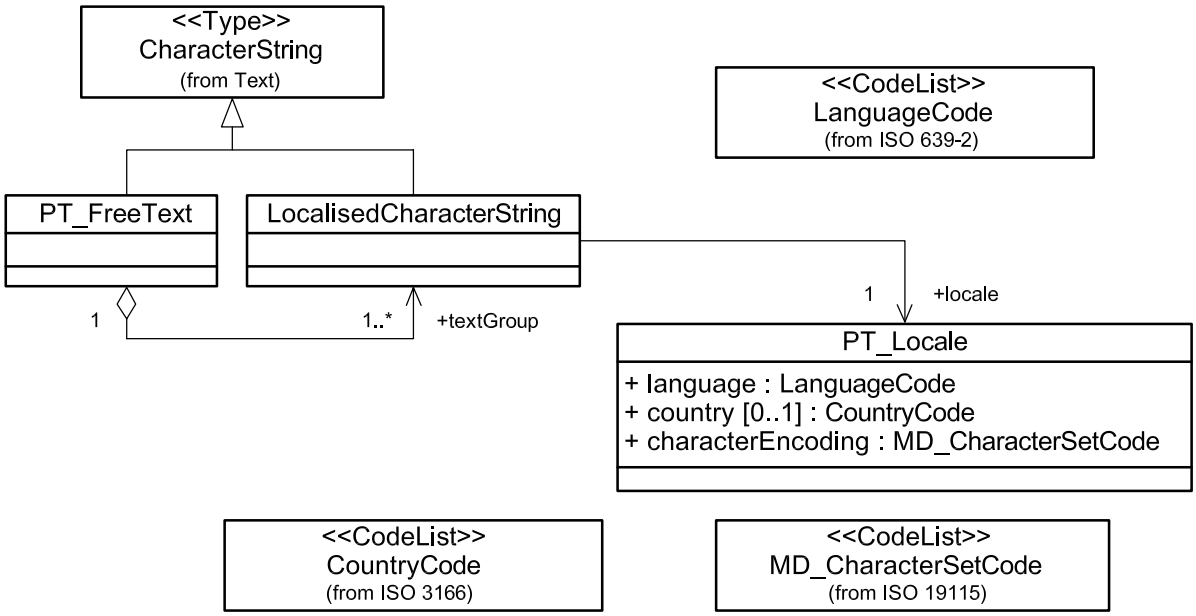

Figure 3 — Free text and localized character string

As a consequence of introducing the Locale concept (PT_Locale), the Group concept (identified as PT_Group) from ISO 19115:2003, Annex J is replaced by the concept of Localised String (identified as LocalisedCharacterString). LocalisedCharacterString is a subtype of CharacterString whose value is expressed in a single locale. An instance of a Free Text consequently is a CharacterString (with its value expressed in a default language and character set that could be defined in an instance of MD_Metadata), which also aggregates a series of Localised Character String translations via the *textGroup* role.

7.3.2 Multilingual metadata sets

An optional but repeatable attribute, locale, presented in Figure 4, was added to the class MD_Metadata of ISO 19115 by corrigendum. This attribute is instantiated if – and only if – the metadata set is multilingual (at least one of the metadata elements is an instance of PT_FreeText or one of its inherited classes).

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

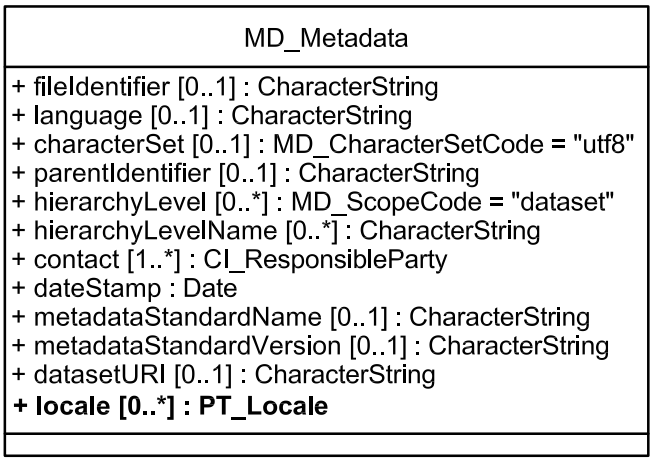

Figure 4 — Attributes of MD_Metadata in the geographic metadata XML schema

7.3.3 Management of localized strings

An instance of free text is composed of default character strings and their translations in different locales through the use of localized strings. This construct implies a distribution of localized strings throughout any given multilingual metadata set. However, a more common way of managing multilingual sets of information consists of grouping the localized strings per their locales. In order to ease the management of localized strings, this Technical Specification describes the concept of locale container (identified as PT_LocaleContainer). A locale container aggregates a set of localized strings related to a given locale (*locale* attribute of PT_LocaleContainer). There is no direct relationship between a locale container and a metadata set except that a locale container may aggregate localized strings of a metadata set.

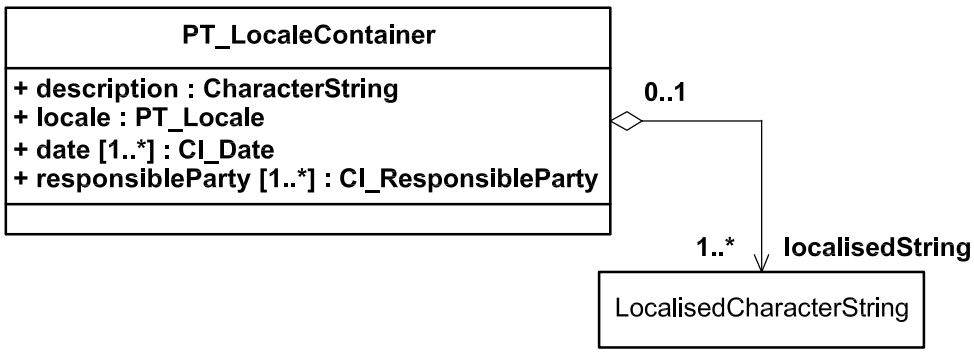

Figure 5 — Translation container

This translation container concept is particularly useful in terms of the XML implementation of ISO 19115, but it is applicable to any other implementation. Indeed, an XML file can only support data expressed in a single character set, which is generally declared in the XML file header. Having all the localized strings stored in a single XML file would limit the use of a single character set such as UTF-8. In order to avoid this,

— the LocalisedCharacterString class is implemented specifically to allow a by-reference containment of the PT_FreeText.textGroup property, and

— the PT_LocaleContainer is the recommended root element to be instantiated in a dedicated XML file.

The localized string related to a given locale can be stored in a corresponding locale container (i.e. XML file) and referenced from the PT_FreeText.textGroup property instances.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

7.4 Extensions for metadata-based transfers of geospatial information

7.4.1 Transfer of datasets and aggregate datasets

Figure 6 reproduces Figure 3 of ISO 19115:2003, focusing on the classes and associations needed for metadata-driven transfers of geospatial information. Datasets (DS_Dataset) may be part of aggregates (DS_Aggregate) which may be subset or superset aggregates. Both aggregates and datasets are linked to one or more sets of metadata elements (MD_Metadata).

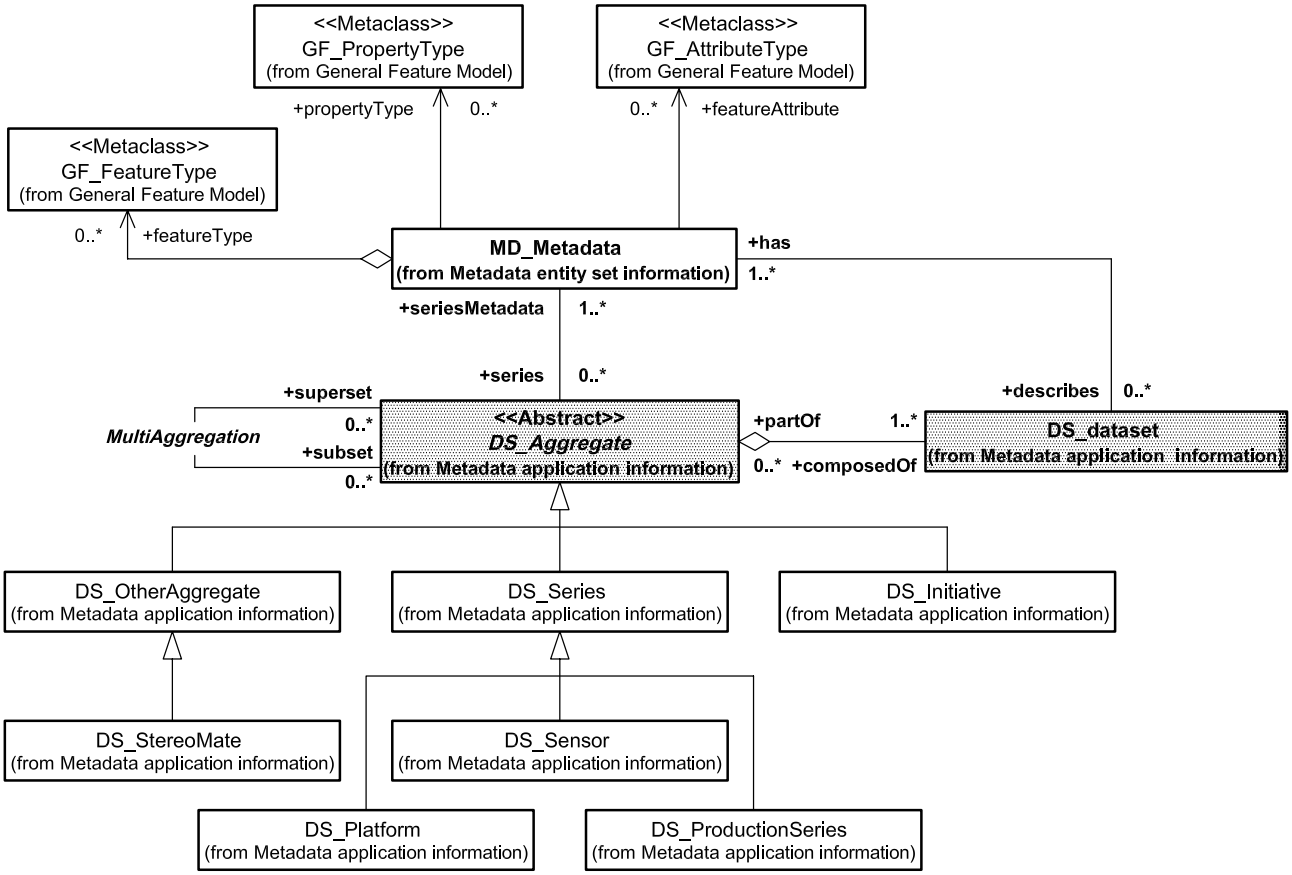

Figure 6 — Datasets, aggregates and their metadata

Metadata-based transfers of geospatial information require the extension of ISO 19115 that is shown in Figure 7.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

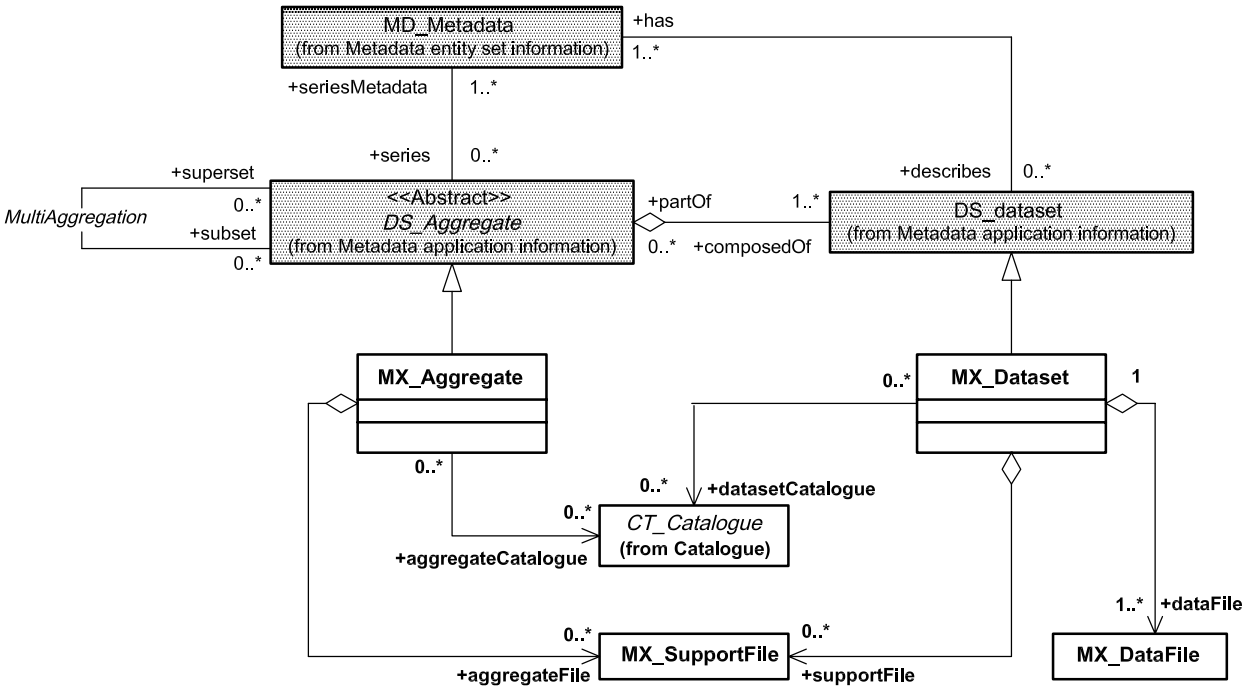

Figure 7 — The generic interchange organization

In order to supply a generic interchange organization for transfer, two new concepts are introduced: **transfer dataset** (MX_Dataset) and **transfer aggregate** (MX_Aggregate).

In the context of an interchange by transfer, the dataset data is organized in data files (MX_DataFile). Both transfer datasets and aggregates may be accompanied by support files (MX_SupportFile) which may contain resources needed to exploit them, or complementary information. The data files and the support files are described in 7.4.3.

In practice, the information needed to exploit a dataset or an aggregate is not limited to their metadata.

Particularly:

— the metadata cites the feature and portrayal catalogues but does not embed them;

— the metadata instances reference information such as codelists, units of measure and coordinate reference systems that all need to be accessed.

All of those resources may be managed externally in on-line registries, but it is usually necessary, in the context of interchange by transfer, to be able to provide that information within the transfer datasets and transfer aggregates. The abstract concept of catalogue (CT_Catalogue detailed in 7.4.4) corresponds exactly to those resources needed to exploit the datasets, aggregates and their metadata. Herein, catalogues are used in the context of interchange by transfer and are associated to transfer datasets (MX_Dataset) and transfer aggregates (MX_Aggregate).

7.4.2 Management of aggregates

The subclassification of DS_Aggregate shown in Figure 6 is fully valid conceptually. Yet, in the context of a data transfer, aggregation of datasets follows constraints (e.g. transfer media capacity) and requirements which outweigh the pure design of geospatial aggregations. The transfer interchange mechanism is based on transfer aggregates, but the initial nature of the geospatial aggregation can be expressed through the hierarchyLevel attribute of MD_Metadata using the extension of the MD_ScopeCode codelist specified in Figure 8.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

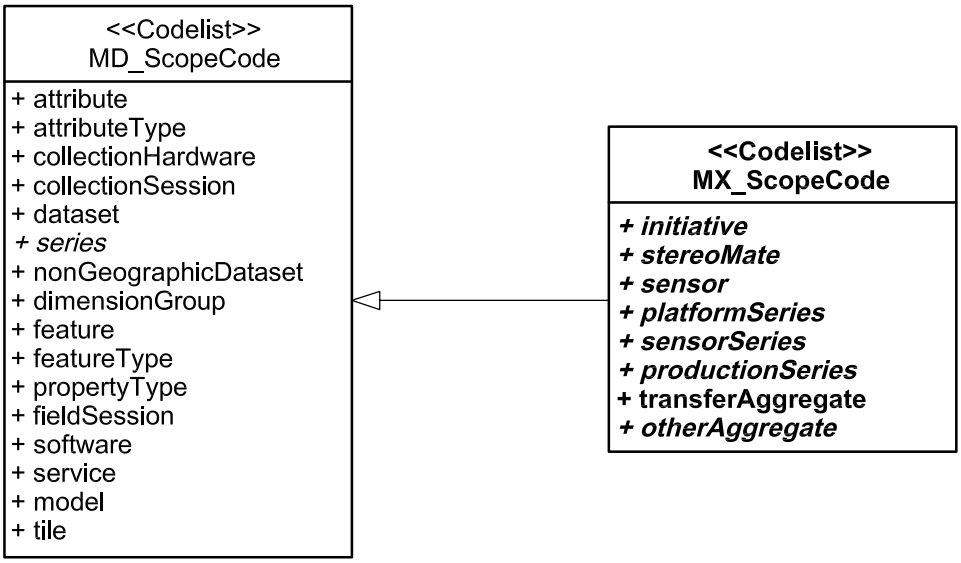

Figure 8 — Extended scope code

Note that the transferAggregate specific value indicates that the aggregate makes sense only in the context of the actual transfer interchange.

7.4.3 Transfer files

Figure 9 describes the support and data files concept. The data files are potentially related to one or many feature types. Each data file has its own format which is generally described within the metadata of the dataset.[3)]

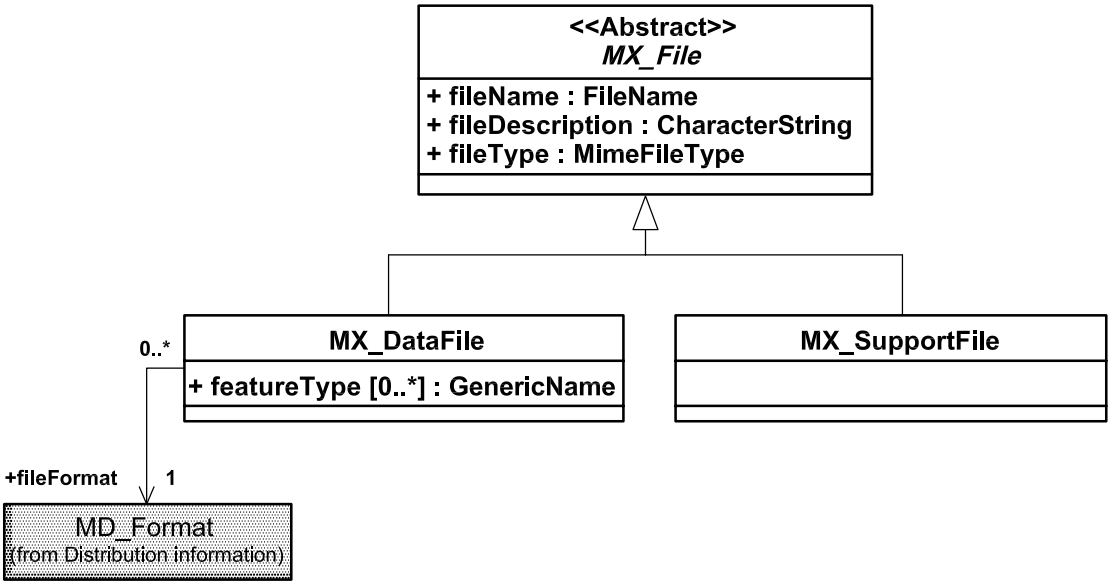

Figure 9 — Transfer files

3) It is also valid to describe the format of data files in the metadata of an aggregate in the case where the format is homogenous over the full aggregate.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

7.4.4 Catalogues

7.4.4.1 General concepts

The abstract concept of catalogue (CT_Catalogue) is shown in Figure 10. In the context of this Technical Specification it is a key component of the standard interchange of geospatial information by transfer. Its design is derived from the feature catalogue concept defined in ISO 19110:

— several optional attributes (language, characterSet and locale) are added to support the cultural and linguistic adaptability requirement;

— several attributes of the class FC_FeatureCatalog are considered specific to a feature catalogue.

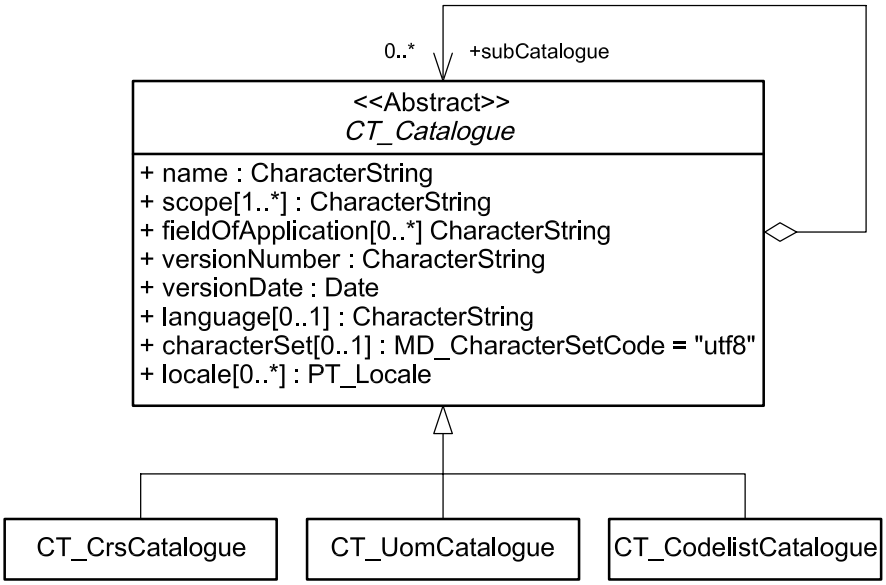

Figure 10 — Catalogues

This Technical Specification provides the conceptual schema of three concrete catalogues as follows.

— A CRS catalogue (CT_CrsCatalogue) contains the description of coordinate reference systems as well as the description of their components (datum, coordinate system, etc.). In the context of interchange by transfer of this specification (detailed in 7.4.1) a CRS catalogue contains the description of items used in the transfer dataset or the transfer aggregate.

— An UoM catalogue (CT_UomCatalogue) contains the description of units of measure. In the context of interchange by transfer of this specification (detailed in 7.4.1) a UoM catalogue contains the description of items used in the transfer dataset or the transfer aggregate.

— A codelist catalogue (CT_CodelistCatalogue) contains the description of codelists, including their name, and the definition as well as the description of their values. In the context of interchange by transfer of this specification (detailed in 7.4.1) a codelist catalogue contains the description of items used in the transfer dataset or the transfer aggregate.

Detailed descriptions of these concrete catalogues are provided respectively in 7.4.4.2 to 7.4.4.4.

The abstract concept of catalogue has also been defined as the basis for harmonization of the different ISO 19100 series catalogue concepts, such as PF_PortrayalCatalogue (ISO 19117) and FC_FeatureCatalogue (ISO 19110). The XML schema implementations of ISO 19110 and ISO 19117 are out

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

of the scope of this Technical Specification, but it is expected that the XML feature and portrayal catalogues will be transferable as their concepts (PF_PortrayalCatalogue and FC_FeatureCatalogue) are inherited from CT_Catalogue.

The abstract concept of a catalogue can also serve as a basis for establishing a data quality measures catalogue (e.g. as ISO 19138). This would allow a quality measure catalogue to be transferred as part of the transfer datasets and aggregates defined herein. More generally, the extension of the catalogue concept and the use of catalogues is recommended for any set of information which is managed or is manageable in on-line registries, when the catalogues enable the exploitation of the transfer datasets and aggregates.

7.4.4.2 UoM catalogue

A unit of measure catalogue (CT_UomCatalogue) as shown in Figure 11 aggregates a list of unit of measure items. Those items are defined in ISO 19136.

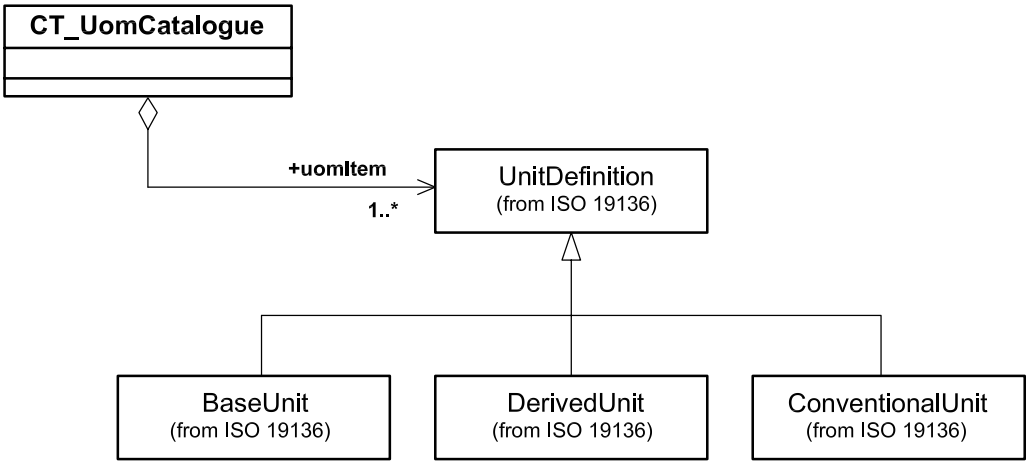

Figure 11 — UoM catalogue

7.4.4.3 CRS catalogue

The concept of CRS catalogue (CT_CrsCatalogue) is shown in Figure 12. A CRS Catalogue has no specific attributes, but it aggregates different types of geodetic items, all inherited from the fundamental classes defined in ISO 19111.

7.4.4.4 Codelist catalogue

The concept of codelist catalogue (CT_CodelistCatalogue) is shown in Figure 13. A codelist catalogue has no specific attributes, but it aggregates a single type of codelist item (CT_Codelist) based on the general concept of catalogue item. A codelist item also aggregates a single type of item (CT_CodelistValue) which corresponds to the values of the codelist.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

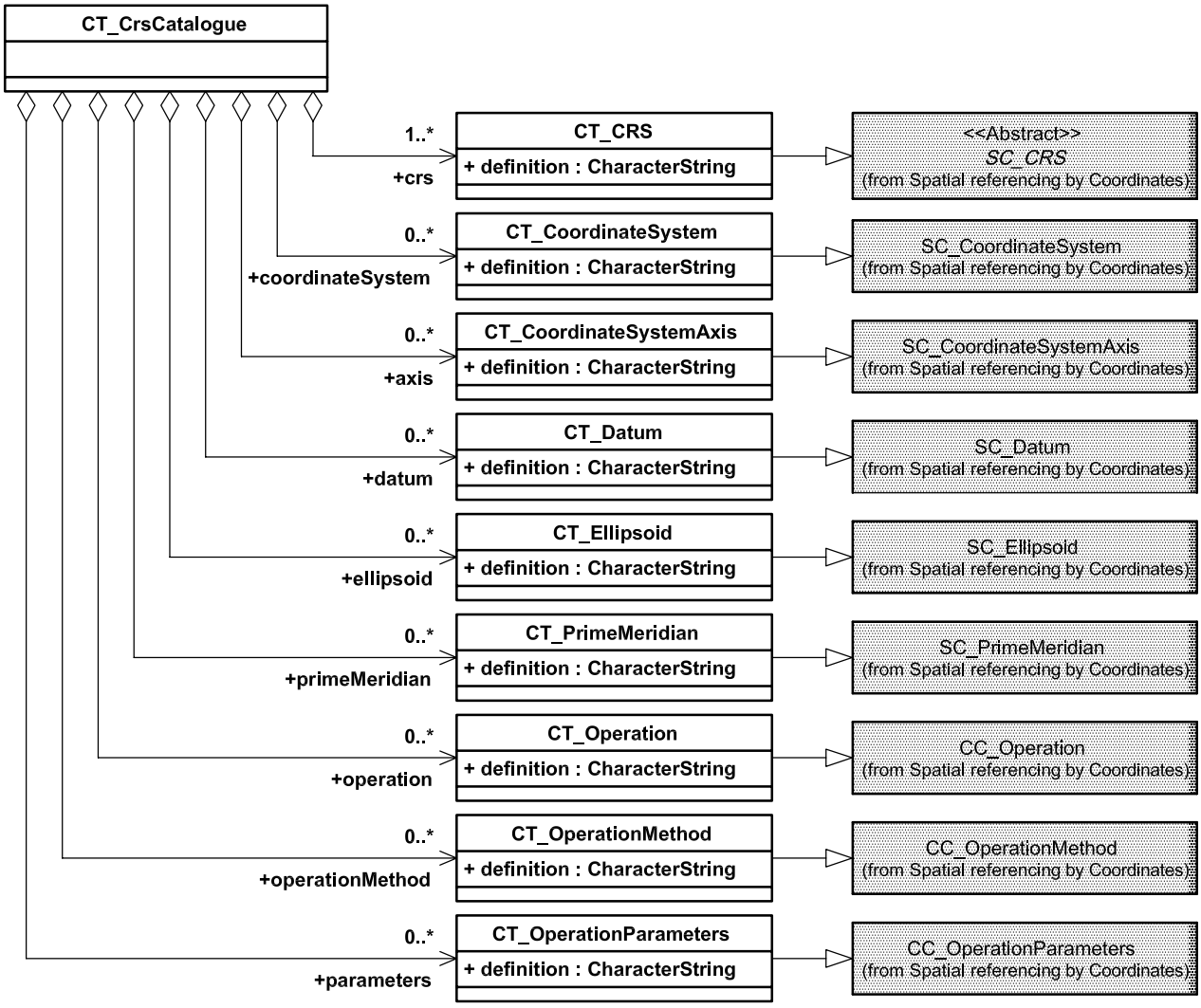

Figure 12 — CRS catalogue

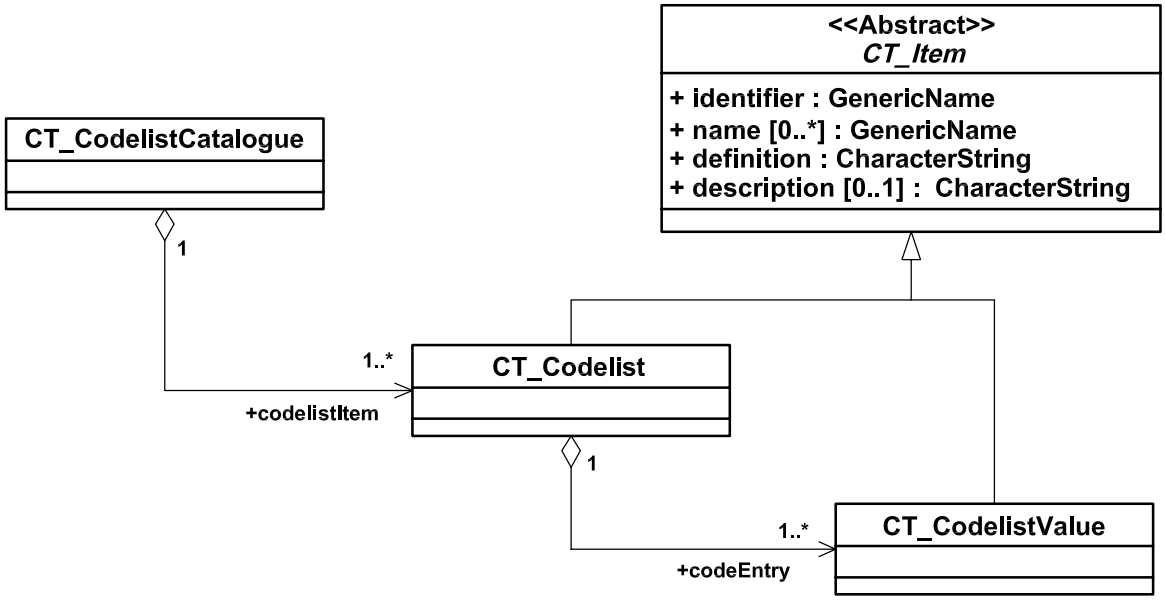

Figure 13 — Codelist catalogue

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

The concept of catalogue item (CT_Item) is introduced to simplify the specification of the components of a codelist catalogue, but its use is highly recommended for the definition of the components of any new type of catalogue.

8 Encoding rules

8.1 Introduction to encoding rules

General rules for transforming UML to XML schema are described in 8.2 to 8.7 and are in accordance with the rules defined in ISO 19118. In some cases ISO 19118 allows for multiple methods of transforming UML to XML schema, and these serve to clarify which method was selected for this implementation specification. Background on classes and how they are the building blocks for encoding all data exchange (and in this case metadata exchange) is purposefully absent from this Technical Specification since ISO 19118 covers this in great detail. This Technical Specification is based on ISO 19118's encoding rules and a familiarity with ISO 19118 will greatly enhance comprehension of the topics described throughout Clause 8.

A default rule is applied to every element of the specified type in the UML profile unless there are special rules for that element as documented in Clause 9. Clause 8 also shows the details of the XML specific packaging of gmd and its associated namespaces while the generic rules for packaging are described in 8.6 and 8.7.

NOTE The way the encoding rules are described does not prevent adopting the best practices in generating the XML schemas.

8.2 Default XML Class Type encoding

ISO 19118 states that the fundamental modelling concept in UML is the class. As a result, the fundamental encoding rules focus on the encoding of a UML class and build from there. It is important to recall from ISO 19118 that a property represents a name-value pair. It can represent an attribute, association, aggregation or composition (see ISO 19118). A class is made up of one or more properties. For example, in Figure 14, the *Class1* class has three properties: *attr1*, *attr2* and *role1*. For the sake of encoding into XML schema it is important to understand that there is no distinction between properties that are UML attributes, associations, aggregations or compositions.

ISO 19118 also describes the need to have identifiers in XML schema and prescribes using identifiers (ids) and universal unique identifiers (uuids) to accomplish this. There is a special XML schema type in the gco namespace, gco: AbstractObject_Type (described in detail in 9.7), which provides the necessary identifiers. It is mentioned here because it is part of the default XML class type encoding.

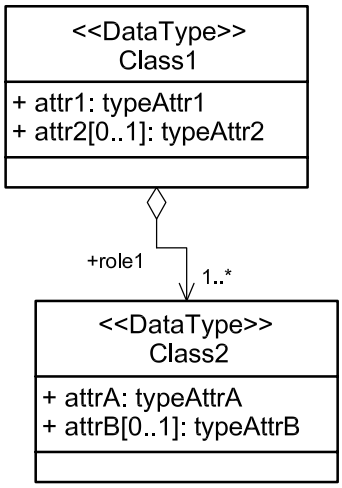

Figure 14 — Sample UML

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

The following list details the steps for encoding a default class.

1) A UML class is encoded into XML schema as an XML complex type: xs:complexType. This XML complex type is hereafter referred to as the XML Class Type (XCT). In XML schema each XCT has a name attribute whose value is the class name with the suffix _Type.

EXAMPLE 1 Step 1) of building the Class1 class shown in Figure 14 is:

```
<xs:complexType name="Class1_Type">
        (...)
</xs:complexType>
```

2) All UML classes following the default encoding rules will have complex content, and to provide this capability, the xs:complexType element contains an xs:complexContent element.

EXAMPLE 2 Step 2) of building the Class1 class shown in Figure 14 is:

```
<xs:complexType name="Class1_Type">
  <xs:complexContent>
        (...)
  </xs:complexContent>
</xs:complexType>
```

3) All UML classes following the default encoding rules will extend the gco:AbstractObject_Type which is done by adding an *xs:extension* element with the base attribute equal to gco:AbstractObject_Type

EXAMPLE 3 Step 3) of building the Class1 class shown in Figure 14 is:

```
<xs:complexType name="Class1_Type">
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
        (...)
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

4) All UML classes following the default encoding rules will have a sequence containing all the properties of the class. This is accomplished by adding an *xs:sequence* element, containing xs:element elements for each property of the class. The attributes of the *xs:element* element will be:

i) the *name* attribute is equal to the name of the property;

ii) the *type* attribute is equal to the name of the XCPT corresponding to the UML class specified as the type of the particular property. Subclause 8.4 explains that by default this is the class name plus "_PropertyType" and 8.5.7 and 8.5.8.2 define the only exceptions to this XCPT naming convention. The name of the XCPT used as the value for this type of attribute will also be properly prefixed with the appropriate namespace as described in 8.6;

iii) the *minOccurs* and *maxOccurs* attributes have the values described in Table A.5 of ISO 19118:2005. Additionally, if a property of the class happens to be an attribute that uses the ‘set’ or ‘sequence’ structure then the *minOccurs* attribute is “0” for optional attributes and “1” for mandatory attributes and the *maxOccurs* is “unbounded”.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

EXAMPLE 4 Step 4) of building the Class1 class shown in Figure 14 is:

```
<xs:complexType name="Class1_Type">
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
      <xs:sequence>
        <xs:element name="attr1" type="ns1:typeAttr1_PropertyType"/>
        <xs:element name="attr2" type="ns1:typeAttr2_PropertyType"
minOccurs="0" />
         <xs:element name="role1" type="ns1:Class2_PropertyType"
minOccurs="1"     maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

NOTE Throughout the examples in this Technical Specification there is a namespace prefix, “ns1” that is not always shown in the UML. This is a fictitious namespace used to illustrate when a namespace prefix should be present but it does not correspond to any of the namespaces or corresponding prefixes defined in this Technical Specification.

The order of the xs:elements within the xs:sequence is based on the order of a property’s corresponding entry in the data dictionary (Annex B) of ISO 19115:2003. Properties are listed in the sequence starting from those with the lowest row number (shown in the first column of all tables in ISO 19115, Annex B) followed by those with a higher row number. If a property of a class is absent from ISO 19115:2003, Annex B, then it occurs following the properties present in ISO 19115:2003, Annex B and based on the order of their presence in the data dictionary of this Technical Specification (Annex B). This ordering rule applies to default encodings as well as all special case encodings for XCTs.

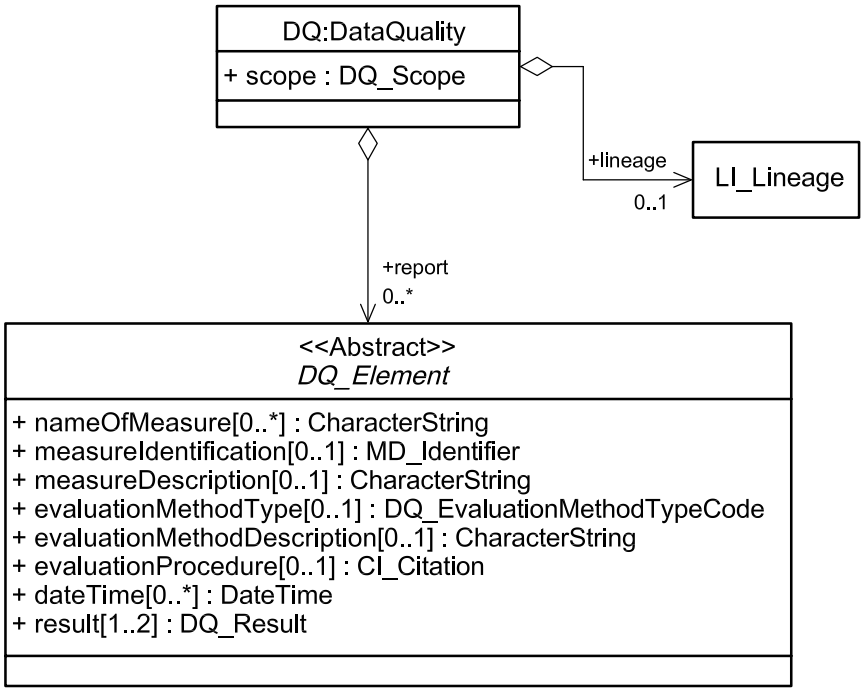

Figure 15 — Partial data quality information UML from ISO 19115:2003, A.2.4.1

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

The XCT corresponding to Figure 15:

```
<xs:complexType name="DQ_DataQuality_Type">
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
      <xs:sequence>
        <xs:element name="scope" type="gmd:DQ_Scope_PropertyType"/>
        <xs:element name="lineage" type="gmd:LI_Lineage_PropertyType"
minOccurs="0"/>
        <xs:element name="report" type="gmd:_DQ_Element_PropertyType"
minOccurs="0"     maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType
```

8.3 XML Class Global Element encoding

Subsequently, a UML class is also encoded as a global element with a name attribute equal to the name of the UML class and the *type* equal to the name of the XCT described in 8.2 (with the appropriate namespace prefix included). Hereafter, this global element is referred to as the XML Class Global Element (XCGE).

EXAMPLE 1 The XCGE of the Class1 class shown in Figure 15 is:

```
<!-- ..................................................... -->
<xs:element name="Class1" type="ns1:Class1_Type"/>
 <!-- ..................................................... -->
```

EXAMPLE 2 The XCGE corresponding to DQ_DataQuality shown in Figure 15 is:

```
<!-- ..................................................... -->
<xs:element name="DQ_DataQuality" type="gmd:DQ_DataQuality_Type"/>
 <!-- ..................................................... -->
```

8.4 XML Class Property Type encoding

8.4.1 Property type encoding for complexType XCTs

To support the property concept described in 8.2 and the potential for a given class to be a property type for a set of container classes, each UML class is also defined as another XSD xs:complexType which is hereafter referred to as the XML Class Property Type (XCPT). Recognise that the containment of a property is managed through the XCPT of its data type. By default both “by Value” and “by Ref” containment are allowed by the XCPT. The following list details the steps for encoding an XCPT for a default class.

1) To distinguish the XCPTs from the XCTs, the xs:complexType element has a name attribute equal to the class name suffixed with _PropertyType.

EXAMPLE 1 Step 1) of building the Class1 class XCPT shown in Figure 14 is:

```
<xs:complexType name="Class1_PropertyType">
    (...)
</xs:complextype>
```

2) To provide “by Value” containment in a manner that supports the polymorphism requirements described in 6.7 the XCPT has an optional xs:sequence element that contains one xs:element element with a ref attribute equal to the XCT of the UML. The xs:sequence element is made optional in order to allow for the possibility that the property will be implemented “by Ref”.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

EXAMPLE 2 Step 2) of building the Class1 class XCPT shown in Figure 14 is:

```
<xs:complexType name="Class1_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="ns1:Class1"/>
  </xs:sequence>
</xs:complexType>
```

3) To provide "by Ref" containment the XCPT has an xs:attributeGroup element with a ref attribute equal to "gco:ObjectReference". The IM_ObjectReference attribute Group is introduced in ISO 19118 and further detailed in 9.7.

EXAMPLE 3 Step 3) of building the Class1 class XCPT shown in Figure 14 is:

```
<xs:complexType name="Class1_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="ns1:Class1"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
</xs:complexType>
```

4) To provide null value explanations when necessary, the XCPT has an xs:attribute element with a ref attribute equal to "gco:nilReason". The nilReason attribute is described in detail in 9.7.3.4.

EXAMPLE 4 Step 4) of building the Class1 class XCPT shown in Figure 14 is:

```
<xs:complexType name="Class1_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="ns1:Class1"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason">
</xs:complexType>
```

EXAMPLE 5 Sample XCPT corresponding to DQ_DataQuality shown in Figure 15.

```
<xs:complexType name="DQ_DataQuality_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="gmd:DQ_DataQuality"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

Notice that there is no specific constraint introduced in the XCPT to force the containment information of a property defined in UML. The UML containment instructions shall be managed by external testing applied to the XML metadata. This issue is discussed in more detail in Annex A.

8.4.2 Property type encoding for simpleType XCTs

There are some cases of specific XCT encodings described in 8.5.4 and throughout Clause 9 where the XCT is an *xs:simpleType* instead of an *xs:complexType* as described in 8.2. When an XCT is *simple* the corresponding XCPT does not include the "by Ref" capabilities described in 8.4.1. Preventing the use of referencing on simple types avoids a potential complexity that would exist if all property types could be implemented by reference.

In 8.5.4 the first simple XCT is introduced for the encoding of classes stereotyped *Enumeration*. The XCPT corresponding to simpleType XCTs, such as for an enumeration, does not include the gco:ObjectReference attributeGroup.

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

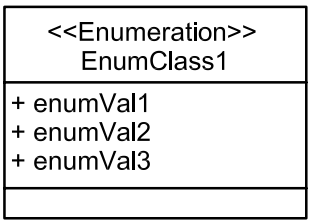

Figure 16 — Sample UML for class stereotyped enumeration (a simpleType)

EXAMPLE The XCPT for the EnumClass1 class shown in Figure 16 is:

```
<xs:complexType name="EnumClass1_PropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="ns1:EnumClass1"/>
    </xs:sequence>
    <xs:attribute ref="gco:nilReason" />
</xs:complexType>
```

Notice that even in the case of simpleTypes there is no specific constraint introduced in the XCPT to force the use of a byVal or a nilReason of a property. This containment has to be managed by external testing applied to the XML metadata. This issue is discussed in more detail in Annex A.

8.5 Special case encodings

8.5.1 Introduction to special case encodings

While the encoding mechanisms described in 8.2 to 8.4 cover the majority of classes found in the UML of the ISO 19100 series of International Standards, there are some special cases where different encoding rules are applied. The exceptions are generally based on special stereotypes applied to classes. A class without a stereotype or a class with the stereotype *Type* or *DataType* will follow the default encodings. Other classes will follow their corresponding encoding rules in 8.5.2 to 8.5.6.

8.5.2 Abstract classes

For implementation purposes it is desirable to better identify abstract class types so that implementers realize that there cannot be elements of such types. Abstract classes in the UML of the ISO 19100 series of International Standards are encoded by:

a) prefixing the word ‘Abstract’ to the abstract class name of the corresponding XCT;

b) adding the *abstract* attribute with a value of “true” to the *xs:complexType* element of the XCT;

c) prefixing the word ‘Abstract’ to the abstract class name of the XCGE;

d) adding the *abstract* attribute with a value of “true” to the *xs:element* element of the XCGE;

e) using the appropriate prefixed class name in the *ref* attribute of the *xs:element* in the corresponding XCPT.

NOTE The word ‘Abstract’ is not prefixed to the name attribute of the XCPT because a property type will not be abstract in XML schema).

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

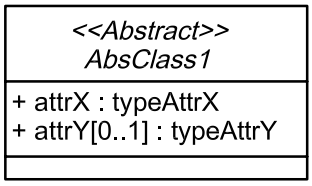

Figure 17 — Sample UML for an abstract class

EXAMPLE 1 The XCT corresponding to AbsClass1 as shown in Figure 17:

```
<xs:complexType name="AbstractAbsClass1_Type" abstract="true">
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
      <xs:sequence>
        <xs:element name="attrX" type="ns1:typeAttrX_PropertyType"/>
       <xs:element name="attrY" type="ns1:typeAttrY_PropertyType"
minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

EXAMPLE 2 The XCGE corresponding to AbsClass1 as shown in Figure 17:

```
<xs:element name="AbstractAbsClass1"
type="ns1:AbstractAbsClass1_Type" abstract="true"/>
```

EXAMPLE 3 The XCPT corresponding to AbsClass1 as shown in Figure 17:

```
<xs:complexType name="AbsClass1_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="ns1:AbstractAbsClass1"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

EXAMPLE 4 Sample XCPT using actual ISO 19115 example.

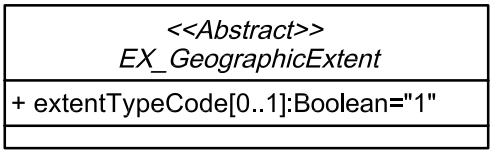

Figure 18 — The UML of the abstract class EX_GeographicExtent from ISO 19115

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

The XCT corresponding to EX_GeographicExtent class shown in Figure 18 is:

```
<xs:complexType name="AbstractEX_GeographicExtent_Type"
abstract="true">
  <xs:complexContent>
    <xs:extension base="gco:AbstractObject_Type">
      <xs:sequence>
        <xs:element name="extentTypeCode"
type="gco:Boolean_PropertyType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The XCGE corresponding to EX_GeographicExtent class shown in Figure 18 is:

```
<xs:element name="AbstractEX_GeographicExtent"
type="gmd:AbstractEX_GeographicExtent_Type" abstract="true"/>
```

The XCPT corresponding to EX_GeographicExtent class shown in Figure 18 is:

```
<xs:complexType name="EX_GeographicExtent_PropertyType">
  <xs:sequence minOccurs="0">
    <xs:element ref="gmd: AbstractEX_GeographicExtent"/>
  </xs:sequence>
  <xs:attributeGroup ref="gco:ObjectReference"/>
  <xs:attribute ref="gco:nilReason"/>
 </xs:complexType>
```

8.5.3 Inheritance and subclass encodings

The concept of inheritance is implemented through generalizations that are considered simple extensions. Admittedly this is a simplification of the concept of inheritance and subclassing since it neglects cases of generalization relationships that provide restriction or cases of generalization relationships that address multiple-inheritance. This extension-only characteristic of generalizations is by design, and the implementation of restrictive generalizations is discussed in A.4. As far as XML schema is concerned, multiple-inheritance is not supported and can only be simulated. Since ISO 19115 has no cases of multiple-inheritance that need to be implemented, mechanisms for the simulation of multiple-inheritance are omitted from these encoding rules.

As with the default UML classes, those that are subclasses have corresponding XCT, XCGE and XCPT in gmd. The XCT of a subclass differs from a default class in the following ways.

a) Within the xs:complexContent element of the XCT there is an *xs:extension* element whose *base* attribute is equal to the namespace qualified XCT of the base (or super) class. (This differs from the XCT of a default UML class *xs:extension* element whose *base* attribute is a gco:AbstractObject_Type to provide id and uuid attributes. Those necessary attributes still exist through the extension of the base class). It is important to note that if the super class is an abstract class, then its corresponding XCT name will be prefixed with the word “Abstract” as described in 8.5.2.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

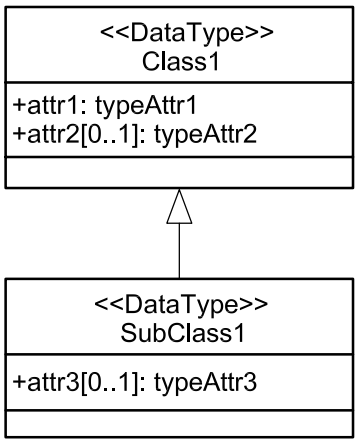

Figure 19 — Sample UML for subclass

EXAMPLE 1 Step a) of building the SubClass1 class shown in Figure 19 is:

```
<xs:complexType name="SubClass1_Type">
    <xs:complexContent>
        <xs:extension base="ns1:Class1_Type">
              (...)
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

b) The *xs:extension* element contains an *xs:sequence* that contains all the properties of the class as described for a default UML class in Step 4) of 8.2. Note that this sequence only contains the properties that are specific to the subclass.

EXAMPLE 2 Step a) of building the SubClass1 class shown in Figure 19 is:

```
<xs:complexType name="SubClass1_Type">
  <xs:complexContent>
    <xs:extension base="ns1:Class1_Type">
      <xs:sequence>
        <xs:element name="attr3" type="ns1:typeAttr3_PropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The only difference between the XCGE of the subclass and the XCGE of a default UML class is that it includes the additional *substitutionGroup* attribute which is equal to the namespace qualified XCGE of the base class.

EXAMPLE 3 The encoding of the XCGE for the SubClass1 class shown in Figure 19 is:

```
<xs:element name="SubClass1" type="ns1:SubClass1_Type"
substitutionGroup="ns1:Class1"/>
```

The XCPT of a subclass is encoded identically to the XCPT of a default UML class.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

EXAMPLE 4 The encoding of the XCPT for the SubClass1 class shown in Figure 19 is:

```
<xs:complexType name="SubClass1_PropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="ns1:SubClass1"/>
    </xs:sequence>
    <xs:attributeGroup ref="gco:ObjectReference"/>
    <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

EXAMPLE 5 Sample SubClass in XML schema using actual ISO 19115 example.

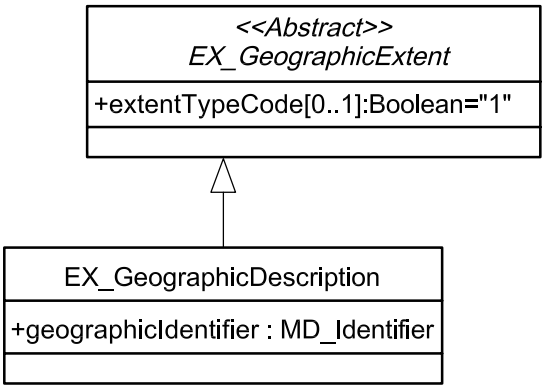

Figure 20 — UML of the classes EX_GeographicExtent and EX_GeographicDescription from ISO 19115

The XCT corresponding to EX_GeographicDescription class shown in Figure 20 is:

```
xs:complexType name="EX_GeographicDescription_Type">
    <xs:complexContent>
        <xs:extension base="gmd:AbstractEX_GeographicExtent_Type">
            <xs:sequence>
                <xs:element name="geographicIdentifier"
type="gmd:MD_Identifier_PropertyType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
 </xs:complexType
```

The XCGE corresponding to EX_GeographicDescription class shown in Figure 20 is:

```
<xs:element name="EX_GeographicDescription"
type="gmd:EX_GeographicDescription_Type"
substitutionGroup="gmd:AbstractEX_GeographicExtent"/>
```

The XCPT corresponding to EX_GeographicDescription class shown in Figure 20 is:

```
<xs:complexType name="EX_GeographicDescription_PropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="gmd:EX_GeographicDescription"/>
    </xs:sequence>
    <xs:attributeGroup ref="gco:ObjectReference"/>
    <xs:attribute ref="gco:nilReason"/>
</xs:complexType
```


Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

8.5.4 Enumeration encodings

The conceptual schema language of the ISO 19100 series of International Standards supports two enumerated types whose "declaration defines a list of valid mnemonic identifiers (see ISO/TS 19103)". The two enumerated types are *Enumeration* and *CodeList*, and it is important to recognise the difference between these two types. A class that is stereotyped *Enumeration* "can contain only simple attributes which represent the enumeration values (see ISO/TS 19103)". An <<Enumeration>> class is used only in the case where no extension to the list of values is necessary, otherwise a <<CodeList>> class is used.

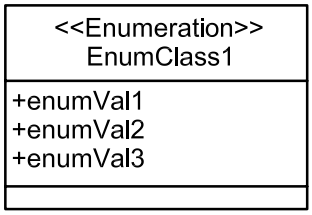

Figure 21 — Sample UML for class stereotyped Enumeration

a) The encoding of an <<Enumeration>> class begins with the creation of an XCT starting with an *xs:simpleType* element with a *name* attribute equal to the class name in the UML with the addition of the _Type suffix.

EXAMPLE 1 Step a) of building the EnumClass1 class shown in Figure 21 is:

```
<xs:simpleType name="EnumClass1_Type">
        (...)
</xs:simpleType>
```

b) The xs:simpleType element contains an xs:restriction element with a *base* attribute equal to an xs:string.

EXAMPLE 2 Step b) of building the EnumClass1 class shown in Figure 21 is

```
<xs:simpleType name="EnumClass1_Type">
  <xs:restriction base="xs:string">
      (...)
  </xs:restriction>
</xs:simpleType>
```

c) Each *xs:restriction* element contains a series of *xs:enumeration* elements whose value attributes equal the attribute values shown in the UML diagram for the enumeration.

EXAMPLE 3 Step c) of building the EnumClass1 class shown in Figure 21 is:

```
<xs:simpleType name="EnumClass1_Type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="enumVal1"/>
        <xs:enumeration value="enumVal2"/>
        <xs:enumeration value="enumVal3"/>
    </xs:restriction>
</xs:simpleType>
```

The XCGE of an <<Enumeration>> class follows the default XCGE rules with the addition of a *substitutionGroup* attribute equal to the "gco:CharacterString" XCGE.

Documents
Document Sharing Hub

Full document details: DD CEN ISO/TS 19139:2009

EXAMPLE 4 The XCGE of the EnumClass1 class shown in Figure 21 is:

```
<xs:element name="EnumClass1" type="ns1:EnumClass1_Type"
substitutionGroup="gco:CharacterString"/>
```

The XCPT of an <<Enumeration>> class follows the default encoding described for simple type XCTs in 8.4.2.

EXAMPLE 5 The XCPT of the EnumClass1 class shown in Figure 21 is:

```
<xs:complexType name="EnumClass1_PropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="ns1:EnumClass1"/>
    </xs:sequence>
    <xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```

EXAMPLE 6 Sample enumeration class using actual ISO 19115 example.

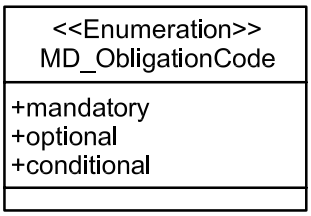

Figure 22 — MD_ObligationCode class from ISO 19115

The XCT corresponding to MD_ObligationCode class shown in Figure 22 is:

```
<xs:simpleType name="MD_ObligationCode_Type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="mandatory"/>
        <xs:enumeration value="optional"/>
        <xs:enumeration value="conditional"/>
    </xs:restriction>
</xs:simpleType>
```

The XCGE corresponding to MD_ObligationCode class shown in Figure 22 is:

```
<xs:element name="MD_ObligationCode"
type="gmd:MD_ObligationCode_Type"
substitutionGroup="gco:CharacterString"/>
```

The XCPT corresponding to MD_ObligationCode class shown in Figure 22 is:

```
<xs:complexType name="MD_ObligationCode_PropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="gmd:MD_ObligationCode"/>
    </xs:sequence>
<xs:attribute ref="gco:nilReason"/>
</xs:complexType>
```


Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

8.5.5 CodeList encodings

8.5.5.1 Building blocks of the CodeList encoding

A class that is stereotyped CodeList is an enumerated type like a class that is stereotyped *Enumeration*. The difference is that the <<CodeList>> class is extensible. All <<CodeList>> classes defined in ISO 19115:2003, Annex B, are described in tables with three columns: Name, DomainCode and Definition.

Table 1 — CI_DateTypeCode <<CodeList>> table from ISO 19115

	Name	Domain code	Definition
1.	CI_DateTypeCode	DateTypCd	identification of when a given event occurred
2.	creation	0001	date identifies when the resource was brought into existence
3.	publication	0002	date identifies when the resource was issued
4.	revision	0003	date identifies when the resource was examined or re-examined and improved or amended

Each row in a CodeList table is intended to describe a unique concept or option that supports the intent of the CodeList's definition. The specific encoding of the CodeList stereotype is meant to support some characteristics intended by the creators of ISO 19115. These characteristics are given below.

a) CodeLists and their associated definitions are controlled in registers. ISO 19115:2003, Annex B is the base concept register of the ISO 19115 CodeLists and is the base source for the establishment of user communities' registers.

b) The Name column of the CodeList tables in ISO 19115:2003, Annex B, contains a value that all software will use to recognise the unique concept or option defined in a row of a CodeList table. As a result, this value is not a proper name in any language, even English.

c) User communities need to express the unique concepts and options of a CodeList using one or many natural languages, dialects or idioms identified here as code spaces. In each code space, a name and possibly a definition can be provided; this name and definition are expressed in a given locale (language, country and character set) when the code space corresponds to a natural language. (This would be like adding additional columns to a CodeList defined in ISO 19115:2003, Annex B.)

d) User communities may need to add new concepts or options that were not considered by the creators of ISO 19115 and it is necessary to control these extensions in user community registers. (This would be like adding additional rows to a CodeList defined in ISO 19115:2003, Annex B.)

To satisfy the intended characteristics of a CodeList, a special "CodeListValue_Type" XCT is created with three attributes:

```
<xs:complexType name="CodeListValue_Type">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="codeList" type="xs:anyURI"  use="required"/>
            <xs:attribute name="codeListValue" type="xs:anyURI"
use="required"/>
            <xs:attribute name="codeSpace" type="xs:anyURI"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The definition of the special XCT makes any CodeList substitutable by a CharacterString. An XCGE based on this special XCT and an XCPT is defined for each CodeList as described in 8.5.5.2. Consequently, any CodeList instance is an XML element. Its name is the name of its XCGE. It handles a value and the three XML attributes defined by the special XCT

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009

The *codeList* attribute contains a URL that references a codeList definition within a registry or a codeList catalogue.

EXAMPLE 1 If Table 1 was contained in a catalogue of codeLists on the ISO/TC 211 web site then a possible value for the *codeList* attribute might be:

codeList = "http://www.tc211.org/ISO19139/resources/codeList.xml#CI_DateTypeCode"

The *codeListValue* attribute carries the identifier of the codeList value definition. This identifier is the value expressed in the Name column of the tables in Annex B of ISO 19115:2003. The codeList catalogue (or registry) is expected to contain an explicit name and definition of the value in the default language of the metadata, as well as alternative expressions in different code spaces, some of them corresponding to the different locales supported by the metadata.

EXAMPLE 2 If it was intended for an instance document to indicate a creation date for a piece of metadata then the value for the codeListValue based on Table 1 would be:

codeListValue="creation"

Each of these alternative expressions is associated to a given codeSpace. The *codeSpace* attribute is an optional identifier (URI); when present it refers to the alternative expression of the codelist value definition effectively expressed as the value of the element. The codeSpace URI for the domain code is **domainCode**.

EXAMPLE 3 If it was intended for an instance document to indicate a creation date for a piece of metadata but the implemented wanted to express this value using domain codes, then the value for the codeSpace based on Table 1 would be:

codeSpace="domainCode"

The content of the element is either the name of the codelist value in the given code space or the name corresponding to the codelist value in the default language of the metadata when the *codeSpace* attribute is not present.

EXAMPLE 4 Here is a full instance of CI_DateTypeCode expressed in the domaineCode code space:

<CI_DateTypeCode codeList= “http://www.tc211.org/ISO19139/resources/codeList.xml#CI_DateTypeCode” codeListValue="creation" codeSpace="domainCode">0001</CI_DateTypeCode>

EXAMPLE 5 Here is a full instance of CI_DateTypeCode expressed in the default language of the metadata (e.g. English):

*<CI_DateTypeCode codeList=“http://www.tc211.org/ISO19139/resources/codeList.xml#CI_DateTypeCode” codeListValue="creation">**Creation**</CI_DateTypeCode>*

The XML element value ensures the user of access to a valid default expression of the actual value of the CodeList, while the three attributes ensure the application the ability access the full definition of the CodeList and its values, typically for customization of the user interface (creation of a drop down box providing the registered or catalogued value of the codelist, multilingual and multicultural management).

8.5.5.2 Details of the CodeList encoding

The CodeListValue_Type was introduced in 8.5.5.1, and that complexType serves as the XCT for all <<CodeList>> classes. The XCGE of a <<CodeList>> class is encoded as a global element with a *name* attribute equal to the name of the UML class and the *type* equal to the CodeListValue_Type. Like an <<Enumeration>> class this XCGE also contains a *substitutionGroup* attribute equal to the "gco:CharacterString" XCGE.

Documents
Document Sharing Hub
Full document details: DD CEN ISO/TS 19139:2009