

DIN EN ISO 19143

ICS 35.240.70

**Geoinformation –
Filter Encoding (ISO 19143:2010);
Englische Fassung EN ISO 19143:2012**

Geographic information –
Filter encoding (ISO 19143:2010);
English version EN ISO 19143:2012

Information géographique –
Codage de filtres (ISO 19143:2010);
Version anglaise EN ISO 19143:2012

Gesamtumfang 93 Seiten

Normenausschuss Bauwesen (NABau) im DIN

Nationales Vorwort

Der Text von ISO 19143:2010 wurde vom Technischen Komitee ISO/TC 211 „Geographic information/ Geomatics“ der Internationalen Organisation für Normung (ISO) erarbeitet und als EN ISO 19143:2012 durch das Technische Komitee CEN/TC 287 „Geoinformation“, dessen Sekretariat vom BSI (Vereinigtes Königreich) gehalten wird, übernommen.

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

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

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

Nationaler Anhang NA (informativ)

Begriffe

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

4.1

Attribut

(en: attribute)

⟨XML⟩ Namen-Werte-Paar, das in einem **Element** enthalten ist

[ISO 19136:2007, Definition 4.1.3]

ANMERKUNG In dieser Internationalen Norm ist, sofern nicht anders angegeben, ein Attribut ein XML- Attribut.

4.2

Client

(en: client)

Software-Komponente, die eine **Operation** von einem **Server** aufrufen kann

[ISO 19128:2005, Definition 4.1]

4.3

Koordinate

(en: coordinate)

Zahl in einer Folge von n Zahlen zum Bezeichnen der Lage eines Punktes in einem n -**dimensionalen** Raum

[ISO 19111:2007, Definition 4.5]

4.4

Koordinatenreferenzsystem

(en: coordinate reference system)

Koordinatensystem, das durch ein Datum auf ein Objekt bezogen ist

[ISO 19111:2007, Definition 4.8]

4.5

Koordinatensystem

(en: coordinate system)

Satz mathematischer Regeln, um festzulegen, auf welche Weise **Koordinaten** Punkten zugeordnet werden

[ISO 19111:2007, Definition 4.10]

4.6

Element

(en: element)

⟨XML⟩ elementare Informationseinheit eines XML-Dokuments, die Kind-Elemente, **Attribute** und Zeichen-
daten enthält

[ISO 19136:2007, Definition 4.1.23]

4.7

Feature

(en: feature)

Abstraktion eines Phänomens der realen Welt

[ISO 19101:2002, Definition 4.11]

ANMERKUNG Ein Feature kann als Klasse oder Instanz vorkommen. Die Begriffe „Featureklasse“ oder „Feature-Instanz“ sollten verwendet werden, wenn nur eines von beiden gemeint ist.

4.8

Feature-Identifikator

(en: feature identifier)

Identifikator, der eine **Feature**-Instanz eindeutig bezeichnet

[ISO 19142:2010, Definition 4.8]

4.9

Featurereferenz

(en: feature reference)

Uniform Resource Identifier, der ein **Feature** identifiziert

4.10

Filterfähigkeiten XML

(en: filter capabilities XML)

in XML kodierte Metadaten, die beschreiben, welche in dieser Internationalen Norm definierten **Prädikate** ein System unterstützt

4.11

Filterausdruck

(en: filter expression)

in XML kodierter **Prädikatausdruck**

4.12

Filterausdruckprozessor

(en: filter expression processor)

Komponente eines Systems, die einen **Filterausdruck** verarbeitet

4.13

Funktion

(en: function)

Regel, die jedes **Element** eines Bereichs (Quelle oder Bereich der Funktion) mit einem einzelnen Element in einem anderen Bereich (Zielbereich, Wertebereich oder Wertemenge) verbindet

[ISO 19107:2003, Definition 4.41]

4.14

Schnittstelle

(en: interface)

benannte Menge von **Operationen**, die das Verhalten einer Entität charakterisieren

[ISO 19119:2005, Definition 4.2]

4.15

expliziter Wert

(en: literal value)

konstanter, explizit festgelegter Wert

ANMERKUNG Im Gegensatz dazu steht ein Wert, der durch Lösen einer Ketten von Substitutionen (z. B. einer Variablen) bestimmt wird.

4.16**Verknüpfungsprädikat**

(en: join predicate)

Filterausdruck, der eine oder mehrere Klauseln umfasst, die Eigenschaften von zwei unterschiedlichen Entitätstypen einschränken

ANMERKUNG In dieser Internationalen Norm handelt es sich bei den Entitätstypen um **Ressourcen**-Typen.

4.17**Namensraum**

(en: name space)

Sammlung von Namen, identifiziert durch einen URI-Verweis, die in XML-Dokumenten als **Elementnamen** und **Attributnamen** verwendet werden

[W3C XML Namespaces]

4.18**Operation****Methode**

(en: operation)

Festlegung einer Transformation oder Abfrage, zu deren Ausführung ein Objekt aufgerufen werden kann

[ISO 19119:2005, Definition 4.3]

4.19**Prädikat**

(en: predicate)

Menge von Rechen**operationen**, die auf eine Dateninstanz angewandt werden, die mit wahr oder falsch bewertet

4.20**Prädikatausdruck**

(en: predicate expression)

formale Syntax zum Beschreiben eines **Prädikats**

4.21**Eigenschaft**

(en: property)

Facette oder **Attribut** eines Objekts, referenziert durch einen Namen

4.22**Auftrag****Anfrage, Anforderung**

(en: request)

Aufruf einer **Operation** durch einen **Client**

[ISO 19128:2005, Definition 4.10]

4.23**Ressource**

(en: resource)

Komponente oder Mittel, die bzw. das eine Anforderung erfüllt

[ISO 19115:2003, Definition 4.10]

ANMERKUNG In dieser Internationalen Norm wird davon ausgegangen, dass eine Ressource eine Identität aufweist.

4.24

Antwort

Ergebnis

(en: response)

Ergebnis einer **Operation**, das von einem **Server** an einen **Client** zurückgegeben wird

[ISO 19128:2005, Definition 4.11]

4.25

Service

Dienst

(en: service)

charakteristischer Teil der Funktionalität, der von einer Entität durch **Schnittstellen** bereitgestellt wird

[ISO 19119:2005, Definition 4.1]

4.26

Server

(en: server)

bestimmte Instanz eines **Dienstes**

[ISO 19128:2005, Definition 4.12]

4.27

Tupel

(en: tuple)

geordnete Liste von Werten

[ISO 19136:2007, Definition 4.1.63]

ANMERKUNG In dieser Internationalen Norm handelt es sich bei der geordneten Liste im Allgemeinen um eine endliche Folge von **Ressourcen**.

4.28

Uniform Resource Identifier

URI

(en: uniform resource identifier)

eindeutiger Identifikator für eine **Ressource**, der in Übereinstimmung mit IETF RFC 2396 aufgebaut ist

[ISO 19136:2007, Definition 4.1.65]

ANMERKUNG Die allgemeine Syntax ist <scheme>::<scheme-specified-part>. Die hierarchische Syntax mit einem Namensraum ist <scheme>://<authority><path>?<query>.

English Version

Geographic information —
Filter encoding
(ISO 19143:2010)

Information géographique —
Codage de filtres
(ISO 19143:2010)

Geoinformation —
Filter Encoding
(ISO 19143:2010)

This European Standard was approved by CEN on 9 March 2012.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, 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, Turkey and United Kingdom.



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

Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

Foreword.....	5
Introduction	6
1 Scope	7
2 Conformance.....	7
3 Normative references	8
4 Terms and definitions	9
5 Conventions	12
5.1 Abbreviated terms	12
5.2 UML notation	12
5.3 Use of examples.....	14
5.4 Namespaces	14
5.5 KVP-encoded parameter lists.....	14
5.6 XML Schema fragments	14
6 Query expressions.....	14
6.1 General.....	14
6.2 Abstract query expressions	15
6.3 Ad hoc query expression	15
6.3.1 General considerations	15
6.3.2 XML encoding	16
6.3.3 KVP-encoding	16
7 Filter	19
7.1 General considerations	19
7.2 Encoding.....	20
7.3 Expressions.....	20
7.3.1 General considerations	20
7.3.2 Encoding.....	21
7.4 Value references	21
7.4.1 General considerations	21
7.4.2 Encoding.....	21
7.4.3 Property names in GML	22
7.4.4 XPath expressions.....	22
7.5 Literals	23
7.5.1 General considerations	23
7.5.2 Encoding.....	24
7.6 Functions.....	24
7.6.1 General considerations	24
7.6.2 Encoding.....	24
7.7 Comparison operators	25
7.7.1 General considerations	25
7.7.2 Encoding.....	26
7.7.3 Parameter discussion	27
7.8 Spatial operators.....	28
7.8.1 General considerations	28
7.8.2 Encoding.....	30
7.8.3 Operator semantics	31
7.8.4 Coordinate reference system handling	32
7.9 Temporal operators	32
7.9.1 General considerations	32

7.9.2	Encoding	33
7.9.3	Time zone handling	34
7.10	Logical operators	34
7.10.1	General considerations.....	34
7.10.2	Encoding	35
7.11	Object identifiers	36
7.11.1	General considerations.....	36
7.11.2	Encoding	36
7.12	Extensions	37
7.12.1	General considerations.....	37
7.12.2	Extending filter using the fes:Function element.....	38
7.12.3	Extending filter by adding new elements.....	38
7.13	Filter capabilities	39
7.14	Encoding	41
7.14.1	Capability categories	41
7.14.2	Conformance clause	41
7.14.3	Id capabilities	43
7.14.4	Scalar capabilities	43
7.14.5	Spatial capabilities	44
7.14.6	Temporal capabilities	45
7.14.7	Functions.....	47
7.14.8	Extended capabilities	47
8	Sorting	48
8.1	General considerations.....	48
8.2	Encoding	48
8.3	Exceptions	49
Annex A	(normative) Conformance testing.....	50
A.1	Test cases for query.....	50
A.2	Test cases for ad hoc query	50
A.3	Test cases for functions	50
A.4	Test cases for resource identification.....	50
A.5	Test cases for minimum standard filter	51
A.6	Test cases for standard filter	51
A.7	Test cases for minimum spatial filter	51
A.8	Test cases for spatial filter	51
A.9	Test cases for minimum temporal filter	51
A.10	Test cases for temporal filter	52
A.11	Test cases for version navigation	52
A.12	Test cases for sorting	52
A.13	Test cases for extended operators	52
A.14	Test cases for XPath	52
A.15	Test cases for schema-element() function	52
Annex B	(informative) Filter schema definitions	53
B.1	General considerations.....	53
B.2	Schema files.....	53
B.2.1	expr.xsd	53
B.2.2	filter.xsd.....	53
B.2.3	query.xsd.....	59
B.2.4	sort.xsd.....	60
B.2.5	filterCapabilities.xsd	60
B.2.6	filterAll.xsd	64
Annex C	(informative) Examples	65
C.1	General considerations.....	65
C.2	XPath example	65
C.3	XPath predicate example	68
C.4	XPath schema-element() example	69
C.5	Filter examples	69

C.6 SortBy example..... 75
C.7 Temporal filter example 77
C.8 Filter capabilities examples 79
Annex D (informative) EBNF for XPath subset..... 85
Annex E (informative) Abstract model 86
E.1 Prerequisites 86
E.2 Predicate 86
E.3 Filter 86
E.4 Query..... 86
Bibliography 87

Foreword

The text of ISO 19143:2010 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 EN ISO 19143:2012 by Technical Committee CEN/TC 287 “Geographic Information” the secretariat of which is held by BSI.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by October 2012, and conflicting national standards shall be withdrawn at the latest by October 2012.

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 implement this European Standard: Austria, Belgium, Bulgaria, Croatia, 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, Turkey and the United Kingdom.

Endorsement notice

The text of ISO 19143:2010 has been approved by CEN as a EN ISO 19143:2012 without any modification.

Introduction

Filter encoding was originated within the OGC.

A fundamental operation performed on a set of data or resources is that of querying in order to obtain a subset of the data which contains certain desired information that satisfies some query criteria and which is also, perhaps, sorted in some specified manner.

The term “projection clause” is used to describe an encoding for specifying which subset of resource properties are presented in the response to a query.

The term “filter or selection clause” is used to describe an encoding of predicates which are typically used in query operations to specify how data instances in a source dataset should be filtered to produce a result set. Each data instance in the source set is evaluated using the filter expression. The overall filter expression always evaluates to true or false. If the expression evaluates to true, the data instance satisfies the expression and is marked as being in the result set. If the overall filter expression evaluates to false, the data instance is not in the result set. Thus, the net effect of evaluating a filter expression is a set of data or resource identifiers which satisfy the predicates in the expression.

The term “sorting clause” is used to describe an encoding for specifying how the data in a response is ordered prior to being presented.

Such encodings are considered system neutral because using the numerous XML tools available today, XML encoded projection, selection and sorting clauses can be easily validated, parsed and then transformed into whatever target query language is required to retrieve or modify resources stored in some persistent object store. For example an XML encoded query composed of a projection, selection and sorting clauses can be transformed into a SQL “SELECT ... FROM ... WHERE ... ORDER BY ...” statement to fetch data stored in a SQL-based relational database. Similarly, the same XML encoded query expression can just as easily be transformed into an XQuery expression in order to retrieve data from XML document.

The XML and KVP encodings of projection, selection and sorting clauses described in this International Standard are common components which can be used together or as individually by a number of web services. Any service that requires the ability to query objects from a web-accessible repository can make use of the XML and KVP encodings of a query expression described in this International Standard. For example the GetFeature operation, defined in ISO 19142, uses the elements derived from definitions in this International Standard to encode query expressions.

1 Scope

This International Standard describes an XML and KVP encoding of a system neutral syntax for expressing projections, selection and sorting clauses collectively called a query expression.

These components are modular and intended to be used together or individually by other standards which reference this International Standard.

EXAMPLE 1 ISO 19142 makes use of some or all of these components.

This International Standard defines an abstract component, named `AbstractQueryExpression`, from which other specifications can subclass concrete query elements to implement query operations.

This International Standard also defines an additional abstract query component, named `AbstractAdhocQueryExpresison`, which is derived from `AbstractQueryExpression` and from which other specifications can subclass concrete query elements which follow the following query pattern:

An abstract query element from which service specifications can subclass a concrete query element that implements a query operation that allows a client to specify a list of resource types, an optional projection clause, an optional selection clause, and an optional sorting clause to query a subset of resources that satisfy the selection clause.

This pattern is referred to as an ad hoc query pattern since the server is not aware of the query until it is submitted for processing. This is in contrast to a stored query expression, which is stored and can be invoked by name or identifier.

This International Standard also describes an XML and KVP encoding of a system-neutral representation of a select clause. The XML representation is easily validated, parsed and transformed into a server-specific language required to retrieve or modify object instances stored in some persistent object store.

EXAMPLE 2 An XML encoded filter can be transformed into a WHERE clause for a SQL SELECT statement to fetch data stored in a SQL-based relational database. Similarly, an XML encoded filter expression can be transformed into an XPath or XPointer expression for fetching data from XML documents.

This International Standard defines the XML encoding for the following predicates.

- a) A standard set of logical predicates: and, or and not.
- b) A standard set of comparison predicates: equal to, not equal to, less than, less than or equal to, greater than, greater than or equal to, like, is null and between.
- c) A standard set of spatial predicates: equal, disjoint, touches, within, overlaps, crosses, intersects, contains, within a specified distance, beyond a specified distance and BBOX.
- d) A standard set of temporal predicates: after, before, begins, begun by, contains, during, ends, equals, meets, met by, overlaps and overlapped by.
- e) A predicate to test whether the identifier of an object matches the specified value.

This International Standard defines the XML encoding of metadata that allows a service to declare which conformance classes, predicates, operators, operands and functions it supports. This metadata is referred to as Filter Capabilities.

2 Conformance

Few usage scenarios require the full implementation of this International Standard to work. Therefore, service providers may want to specify requirements for only the subset needed to fulfil their service. Or system developers may want to document which subset of this International Standard it is that they have implemented and conform to. These named conformance classes help in specifying such subsets.

This International Standard defines conformance classes based on the operations and behaviour that a filter encoding service claims to implement. Table 1 indicates which behaviour shall be implemented for each of the conformance classes. The described behaviour shall be implemented for the corresponding conformance class, and the name of the paragraph of the actual detailed abstract test suite in Annex A.

Table 1 — FE conformance classes

Conformance class name	Operation or behaviour	Subclause of the abstract test suite
Query	Service that references this International Standard materializes a concrete query element that is substitutable for fes:AbstractQueryElement.	A.1
Ad hoc Query	Service that references this International Standard materializes a concrete query element that is substitutable for fes:AbstractAdhocQueryElement and materializes a concrete selection clause element that is substitutable for fes:AbstractSelectionClause and materializes a concrete projection clause element that is substitutable for fes:AbstractProjectionClause and materializes a concrete sorting clause element that is substitutable for fes:AbstractSortingClause.	A.2
Functions	Implements functions that are in addition to the operators defined in this International Standard.	A.3
Resource Identification	Implements the ResourceId operator with the rid parameter to allow predicates to be written that allow a specific resource to be queried.	A.4
Minimum Standard Filter	Implements the comparison operators: PropertyIsEqualTo, PropertyIsNotEqualTo, PropertyIsLessThan, PropertyIsGreaterThan, PropertyIsLessThanOrEqualTo, PropertyIsGreaterThanOrEqualTo. Implements the logical operators. Does not implement any additional functions.	A.5
Standard Filter	Implements all the comparison and logical operators and may implement one or more additional functions.	A.6
Minimum Spatial Filter	Implements only the BBOX spatial operator.	A.7
Spatial Filter	Implements the BBOX spatial operator and one or more of the other spatial operators.	A.8
Minimum Temporal Filter	Implements only the During temporal operator.	A.9
Temporal Filter	Implements the During temporal operator and one or more of the other temporal operators.	A.10
Version navigation	Implements ResourceId operator with the parameters that allow versions of resources to be queried (version, startTime, endTime).	A.11
Sorting	Implements sorting of the resources in a response.	A.12
Extended Operators	Implements additional operators not defined in this International Standard.	A.13
Minimum XPath	Implements the minimum required set of XPath capabilities.	A.14
Schema Element Function	Implements the schema-element() XPath function.	A.15

Other standards that include this International Standard shall declare what constitutes a “minimum” filter by declaring the minimum set of conformance classes from Table 1 that shall be implemented.

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 19108:2002, *Geographic information — Temporal schema*

ISO 19125-1:2004, *Geographic information — Simple feature access — Part 1: Common architecture*

ISO 19136:2007, *Geographic information — Geography Markup Language (GML)*

IETF RFC 2396, *Uniform Resource Identifiers (URN): Generic Syntax* (August 1998)

OGC 06-121r3, *OGC Web Services Common Specification*, OGC® Implementation Specification (9 February 2009)

W3C XML, *Extensible Markup Language (XML) 1.0 (Third edition)*, W3C Recommendation (4 February 2004)

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

W3C XML, *Path Language, XML Path Language (XPath) 2.0*, W3C Recommendation (23 January 2007)

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

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

4 Terms and definitions

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

4.1

attribute

⟨XML⟩ name-value pair contained in an **element**

[ISO 19136:2007, definition 4.1.3]

NOTE In this International Standard, an attribute is an XML attribute unless otherwise specified.

4.2

client

software component that can invoke an **operation** from a **server**

[ISO 19128:2005, definition 4.1]

4.3

coordinate

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

[ISO 19111:2007, definition 4.5]

4.4

coordinate reference system

coordinate system that is related to an object by a datum

[ISO 19111:2007, definition 4.8]

4.5

coordinate system

set of mathematical rules for specifying how **coordinates** are to be assigned to points

[ISO 19111:2007, definition 4.10]

4.6

element

⟨XML⟩ basic information item of an XML document containing child elements, **attributes** and character data

[ISO 19136:2007, definition 4.1.23]

4.7

feature

abstraction of real world phenomena

[ISO 19101:2002, definition 4.11]

NOTE A feature can occur as a type or an instance. It is intended that the term “feature type” or “feature instance” be used when only one is meant.

4.8

feature identifier

identifier that uniquely designates a **feature** instance

[ISO 19142:2010, definition 4.8]

4.9

feature reference

Uniform Resource Identifier that identifies a **feature**

4.10

filter capabilities XML

metadata, encoded in XML, that describes which **predicates** defined in this International Standard a system implements

4.11

filter expression

predicate expression encoded using XML

4.12

filter expression processor

component of a system that processes a **filter expression**

4.13

function

rule that associates each **element** from a domain (source, or domain of the function) to a unique element in another domain (target, co-domain, or range)

[ISO 19107:2003, definition 4.41]

4.14

interface

named set of **operations** that characterize the behaviour of an entity

[ISO 19119:2005, definition 4.2]

4.15

literal value

constant, explicitly specified value

NOTE This contrasts with a value that is determined by resolving a chain of substitution (e.g. a variable).

4.16

join predicate

filter expression that includes one or more clauses that constrain properties from two different entity types

NOTE In this International Standard, the entity types are **resource** types.

4.17

namespace

⟨XML⟩ collection of names, identified by a URI reference which are used in XML documents as **element** names and **attribute** names

[W3C XML Namespaces]

4.18

operation

specification of a transformation or query that an object may be called to execute

[ISO 19119:2005, definition 4.3]

4.19

predicate

set of computational **operations** applied to a data instance which evaluate to true or false

4.20

predicate expression

formal syntax for describing a **predicate**

4.21

property

facet or **attribute** of an object referenced by a name

4.22

request

invocation of an **operation** by a **client**

[ISO 19128:2005, definition 4.10]

4.23

resource

asset or means that fulfils a requirement

[ISO 19115:2003, definition 4.10]

NOTE In this International Standard, a resource is assumed to have identity.

4.24

response

result of an **operation** returned from a **server** to a **client**

[ISO 19128:2005, definition 4.11]

4.25

service

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

[ISO 19119:2005, definition 4.1]

4.26

server

particular instance of a **service**

[ISO 19128:2005, definition 4.12]

4.27

tuple

ordered list of values

[ISO 19136:2007, definition 4.1.63]

NOTE In this International Standard, the ordered list is generally a finite sequence of **resources**.

4.28

Uniform Resource Identifier

URI

unique identifier for a **resource**, structured in conformance with IETF RFC 2396

[ISO 19136:2007, definition 4.1.65]

NOTE The general syntax is <scheme>::<scheme-specified-part>. The hierarchical syntax with a namespace is <scheme>://<authority><path>?<query>.

5 Conventions

5.1 Abbreviated terms

BBOX	Bounding Box
CRS	Coordinate Reference System
EBNF	Extended Backus-Naur Form
EPSG	European Petroleum Survey Group
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
KVP	Keyword-value Pair
OGC	Open Geospatial Consortium
SRS	Spatial Reference System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
WFS	Web Feature Service
XML	Extensible Markup Language

5.2 UML notation

5.2.1 Figure 1 describes the Unified Modelling Language (UML) notations used in this International Standard for UML class diagrams.

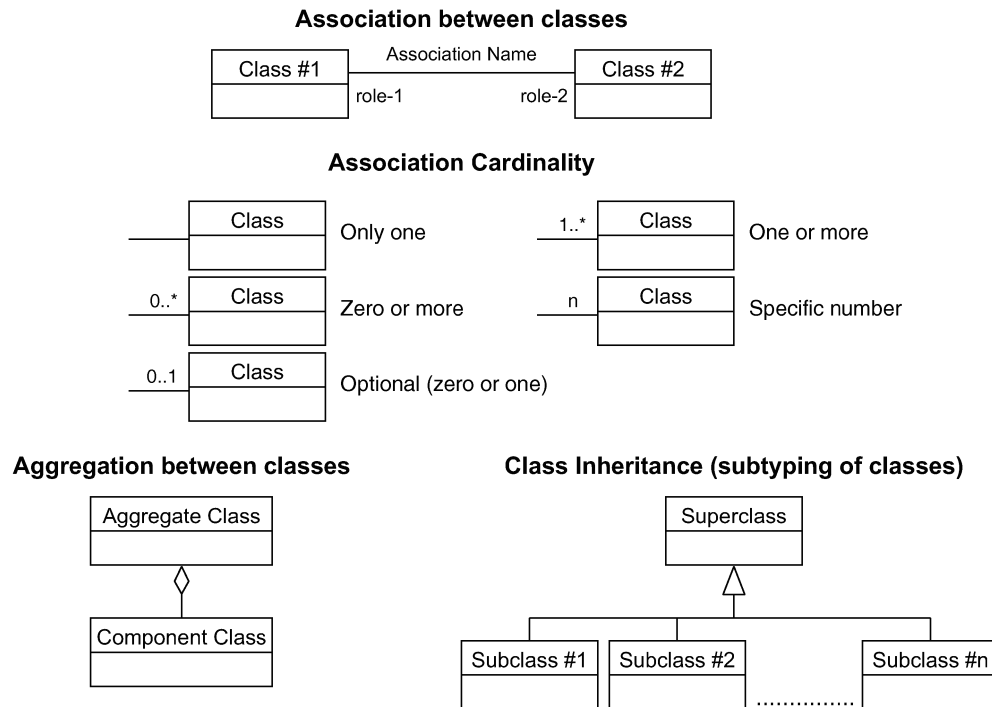


Figure 1 — UML notation in class diagrams

5.2.2 In these class diagrams, the following stereotypes of UML classes are used:

- a) `<<DataType>>` is a descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A `DataType` is a class with no operations, whose primary purpose is to hold the information.
- b) `<<Enumeration>>` is a data type whose instances form a list of alternative literal values. Enumeration means a short list of well-understood potential values within a class.
- c) `<<CodeList>>` is a flexible enumeration for expressing a long list of potential alternative values. If the list alternatives are completely known, an enumeration shall be used; if the only likely alternatives are known, a code list shall be used.
- d) `<<Interface>>` is a definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.
- e) `<<Type>>` is a stereotyped class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A Type class may have attributes and associations.
- f) `<<Union>>` is a list of alternate attributes where only one of those attributes may be present at any time.

See also ISO/TS 19103:2005, 6.8.2 and D.8.3.

5.2.3 In this International Standard, the following standard data types are used:

- a) `CharacterString` is a sequence of characters;
- b) `LocalisedCharacterString` is a `CharacterString` associated with a locale;
- c) `Boolean` is a value specifying `TRUE` or `FALSE`;
- d) `URI` is an identifier of a resource that provides more information;
- e) `Integer` is an integer number.

5.3 Use of examples

This International Standard makes use of XML examples. They are meant to illustrate the various aspects of filters discussed in this International Standard. While every effort has been made to ensure that the examples are well formed and valid, this goal may be sacrificed for the sake of clarity. For instance, many examples are formatted in a specific way to highlight a particular aspect that would render the example invalid from the perspective of an XML validation tool. Furthermore, most examples reference fictitious servers and data.

Thus, this International Standard does not assert that any XML encoded example, copied from this International Standard, would necessarily execute correctly or validate using a particular XML validation tool.

5.4 Namespaces

Namespaces (as specified in W3C XML Namespaces) are used to discriminate XML vocabularies from one another. The following namespaces are normatively used in this International Standard:

- a) (<http://www.opengis.net/fes/2.0>): for the Filter vocabulary;
- b) (<http://www.opengis.net/gml/3.2>): for the GML vocabulary.

5.5 KVP-encoded parameter lists

This International Standard defines both XML and KVP encodings for query and filter expressions. Several of the parameters in the KVP-encoding consist of lists of values (see Table 2) and possibly lists of lists of values. This subclause defines how to encode lists of values as the value of a parameter.

Parameters consisting of lists shall use the comma (",") as the delimiter between items in the list. In addition, multiple lists may be specified as the value of a parameter by enclosing each list in parentheses; "((", ")).

EXAMPLE 1 This example shows a list of items.

PARAMETER=item1,item2,item3,item4a%2Citem4b

This list consists of four values: item1, item2, item3 and the value "item4a,item4b".

NOTE In this example, the embedded comma in the last item has been encoded as per IETF RFC 2396 in order to distinguish it from the commas used in the list of delimited list entries.

EXAMPLE 2 This example shows multiple lists of items assigned to a single parameter.

PARAMETER=(item11,item12,item13)(item21,item22,item23)

5.6 XML Schema fragments

This International Standard makes use of XML Schema (as given in W3C XML Schema Part 1 and W3C XML Schema Part 2) fragments to define the XML encoding of the components of a filter expression. These XML Schema fragments are collected into a set of consolidated schema files in Annex B.

6 Query expressions

6.1 General

A query expression (see Figure 2) is an action that performs a search over some set of resources and returns a subset of those resources. Other standards that reference this International Standard shall assert what a resource is.

EXAMPLE A WFS would assert that a resource is a feature.

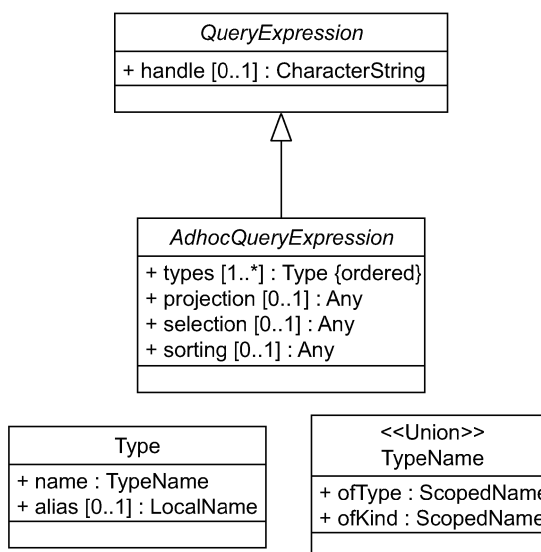


Figure 2 — Query expressions

6.2 Abstract query expressions

This International Standard defines the abstract element `fes:AbstractQueryExpression` as the head of a substitution group of query expressions. The element `fes:AbstractQueryExpression` is defined by the following XML Schema fragment:

```

<xsd:element name="AbstractQueryExpression"
    type="fes:AbstractQueryExpressionType" abstract="true"/>
<xsd:complexType name="AbstractQueryExpressionType" abstract="true">
    <xsd:attribute name="handle" type="xsd:string"/>
</xsd:complexType>

```

The `fes:AbstractQueryExpression` element defines the `handle` attribute which can be used to assign user-defined identifier to the query expression for the purpose of error handling or correlating the response to a query, from within a series of queries, with the source query expression.

International Standards that reference this International Standard shall declare the type(s) of resources that can be queried and shall derive query expressions from `fes:AbstractQueryExpression`.

6.3 Ad hoc query expression

6.3.1 General considerations

A fundamental type of query expression is the ad hoc query expression. It is ad hoc in the sense that the query is not known before the time it is being executed as, for example, a stored query would be.

An ad hoc query expression is a query expression that contains the names of one or more resource types to query, an optional projection clause enumerating the properties of the resource to present in the response, an option selection clause that constraints the properties of those resources types in order to define a result set and an optional sorting clause specifying the order in which the result set is presented.

This subclause defines the head of an substitution group called `fes:AbstractAdhocQueryExpression` from which standards that reference this International Standard can derive concrete ad hoc query expressions.

6.3.2 XML encoding

The following XML Schema fragment define the abstract element fes:AbstractAdhocQueryExpression

```
<xsd:element name="AbstractAdhocQueryExpression"
  type="fes:AbstractAdhocQueryExpressionType"
  substitutionGroup="fes:AbstractQueryExpression"
  abstract="true"/>
<xsd:complexType name="AbstractAdhocQueryExpressionType" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="fes:AbstractQueryExpressionType">
      <xsd:sequence>
        <xsd:element ref="fes:AbstractProjectionClause"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="fes:AbstractSelectionClause" minOccurs="0"/>
        <xsd:element ref="fes:AbstractSortingClause" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="typeNames"
        type="fes:TypeNamesListType" use="required"/>
      <xsd:attribute name="aliases"
        type="fes:AliasesType"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="TypeNamesListType">
  <xsd:list itemType="fes:TypeNamesType"/>
</xsd:simpleType>
<xsd:simpleType name="TypeNamesType">
  <xsd:union memberTypes="fes:SchemaElement xsd:QName"/>
</xsd:simpleType>
<xsd:simpleType name="SchemaElement">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="schema\-element\(.+\)"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AliasesType">
  <xsd:list itemType="xsd:NCName"/>
</xsd:simpleType>
<xsd:element name="AbstractProjectionClause" abstract="true"/>
<xsd:complexType name="AbstractProjectionClauseType" abstract="true"/>
<xsd:element name="AbstractSelectionClause" abstract="true"/>
<xsd:complexType name="AbstractSelectionClauseType" abstract="true"/>
<xsd:element name="AbstractSortingClause" abstract="true"/>
<xsd:complexType name="AbstractSortingClauseType" abstract="true"/>
```

6.3.3 KVP-encoding

Table 2 defines the KVP-encoding for an ad hoc query expression.

Table 2 — KVP-encoding for ad hoc query expression

URL Component	O/M ^a	Description
TYPENAMES	M ^b	A comma-separated list of resource types to query. Specifying more than one name indicates that a join is being performed.
ALIASES	O	A comma-separated list of aliases for the resource types listed as the value of the TYPENAMES parameter.
Projection clause		
PROPERTYNAME	O	If more than one feature type name is specified as the value of the TYPENAMES keyword, a list of parameter lists shall be specified (see 5.5.). Each sublist shall correspond 1:1 with each feature type name listed as the value of the TYPENAMES parameter.
Selection clause		
FILTER (Mutually exclusive with RESOURCEID and BBOX)	O	The value of the parameter shall be a filter expression encoded using the language specified by the FILTER_LANGUAGE parameter.
FILTER_LANGUAGE	O	Indicates the predicate language used to encode the filter expression that is the value of the FILTER parameter. The default value urn:ogc:def:query:Language:OGC-FES:Filter shall be used to indicate that the value of FILTER parameter is a string encoding the filter using an XML fragment as defined in this International Standard.
RESOURCEID (Mutually exclusive with FILTER and BBOX)	O	A comma-separated list of resource identifiers to retrieve from some data store.
BBOX (Mutually exclusive with FILTER and RESOURCEID)	O	A bounding rectangle, encoded as specified in OGC 06-131r3, indicating that all resources which intersect that BBOX shall be retrieved from some data store.
Sorting clause		
SORTBY	O	The SORTBY parameter is used to specify a list of value references that should be used to order (upon presentation) the set of resource instances that satisfy the query. The value of the SORTBY parameter shall have the form "PropertyName [ASC DESC][,PropertyName [ASC DESC],...]" where the letters ASC are used to indicate an ascending sort and the letters DESC are used to indicate a descending sort. If neither ASC nor DESC are specified, the default sort order shall be ascending. An example value might be: "SORTBY=Field1 DESC,Field2 DESC,Field3". In this case the results are sorted by Field 1 descending, Field2 descending and Field3 ascending.
^a O = Optional, M = Mandatory. ^b Standards that reference this International Standard may change the requirement for the TYPENAME parameter. In such cases, the referencing standard shall document whether the TYPENAME parameter is mandatory, optional or mandatory in some cases and optional in others.		

6.3.3.1 Parameter discussion

6.3.3.1.1 typeNames parameter

The mandatory typeNames parameter shall be used within an ad hoc query expression to encode the names of one or more correlated resource types to be queried. Individual resource type names shall be encoded as QName (as given in W3C XML Schema Part 2).

NOTE For the KVP-encoding of the typeNames parameter, see 5.5 for encoding lists of values.

International Standards that derive query expressions from fes:AbstractQueryExpression shall declare the resource type(s) that can be queried.

6.3.3.1.2 aliases parameter

The optional aliases parameter may be used within an ad hoc query expression to specify alternate names for the resource type names specified as the value of the typeName parameter. A resource type alias may be used anywhere; the resource type name may be used within the context of the query expression.

The number of list elements in the value of the aliases parameter shall match the number of corresponding resource type names in the value of the typeName parameter and shall be correlated 1:1.

EXAMPLE 1 < ... typeName="ns1:ResourceType1, ns2:ResourceType2" aliases="A B" ...>

This example encodes an ad hoc query expression fragment that queries the resource types ns1:ResourceType1 and ns2:ResourceType2 which are aliased to A and B. Thus, the tokens A or B can be used within the filter expression of the query expression as alternate names for the resource types ns1:ResourceType1 and ns2:ResourceType2.

Each alias specified in the value of aliases attribute shall be unique within the context of a single query expression.

If the aliases attribute is used, an alias shall be specified for each resource type name listed as the value of typeName attribute.

Aliases are typically used in query expressions that perform a join operation to support self-joins. That is a join of one resource type back to itself.

EXAMPLE 2 typeName="myns:ResType1 myns:ResType1" aliases="a b"

In this example, the first resource type, myns:ResType1, is aliased to the name "a" and the second resource type, myns:Feat1, is aliased to the name "b". Thus properties from the first instance of myns:ResType1 can be referenced in a request as "/a/myns:property_name" and properties from the second instance of myns:ResType2 can be referenced in a request as "/b/myns:property_name" where the token "myns:property_name" is used as a place holder for the name of any property of the resource type myns:ResType1.

6.3.3.1.3 Projection clause

A projection clause encodes a list of optional resource properties that shall be available in a query response.

For XML-encoded requests, specifications which implement the projection clause of an ad hoc query expression shall define a concrete element derived from fes:AbstractAdhocProjectionClause.

For KVP-encoded requests, the PROPERTYNAME keyword shall be used to encode the projection clause. The value of the PROPERTYNAME keyword shall be a comma-separated list of property names.

6.3.3.1.4 Selection clause

The selection clause defines a set of query predicates that shall be applied to a dataset in order to define a subset of data to be operated upon.

Services that implement this International Standard shall use the fes:Filter element, which is substitutable for fes:AbstractSelectionClause, to encode the selection clause of a query expression.

For XML-encoded requests, the selection clauses shall be encoded using the fes:Filter element.

For KVP-encoded requests, the select clause shall be encoded using the keywords FILTER, FILTER_LANGUAGE, RESOURCEID, BBOX (see Table 2). The parameters FILTER, RESOURCEID and BBOX are mutually exclusive. In the event that a selection clause specifies more than one of these parameters, an OperationNotSupported (as given in OGC 06-121r3, Table 25) exception shall be raised.

6.3.3.1.5 Join queries

A join query finds tuples (i.e. pairs, triples, etc.) of resources, among a list of resource types, that satisfy a filter expression which includes join predicates. If the filter expression is satisfied, that tuple of resources is considered to be in the result set of the query expression.

A join query is encoded by:

- a) listing the resource types to join using the `typeNames` parameter (see 6.3.3.1.1);
- b) specifying join predicates in the selection clause that reference properties of the resource types listed as the values of the `typeNames` parameter (see 6.3.3.1.1).

Services that implement join queries shall implement an inner join meaning that only resource tuples which match the join conditions shall be returned in the result set.

6.3.3.1.6 `schema-element()` function

If the list of values for the `typeNames` parameters contains a single QName then the `schema-element()` function can be used to trigger a sequence of queries on the specified resource type and any resource type whose object elements are in the substitution group of the specified resource type.

EXAMPLE `typeNames="schema-element(ns1:Vehicles)"` might, along with `ns1:Vehicle`, query the resource types `ns1:Cars`, `ns1:Boats`, etc ...

The `schema-element()` function shall not be used if a join operation (see 6.3.3.1.5) is being performed.

6.3.3.1.7 Sorting clause

A sorting clause can be used to assert the order in which resources shall appear in response to an ad hoc query expression.

For XML-encoded requests, standards that reference this International Standard shall use the `fes:SortBy` element (see Clause 9), which is substitutable for `fes:AbstractSortingClause`, to encode the sorting clause of an ad hoc query expression.

For KVP-encoded requests, the keyword `SORTBY` shall be used to encode a sorting clause (see Table 2).

7 Filter

7.1 General considerations

A filter (see Figure 3) is used to identify a subset of resources from a collection of resources whose property values satisfy a set of logically connected predicates. If the property values of a resource satisfy all the predicates in a filter then that resource is considered to be part of the resulting subset.

This clause defines the XML encoding of a filter as a set of predicate expressions, contained within the root element `fes:Filter`, encoded using the elements defined herein.

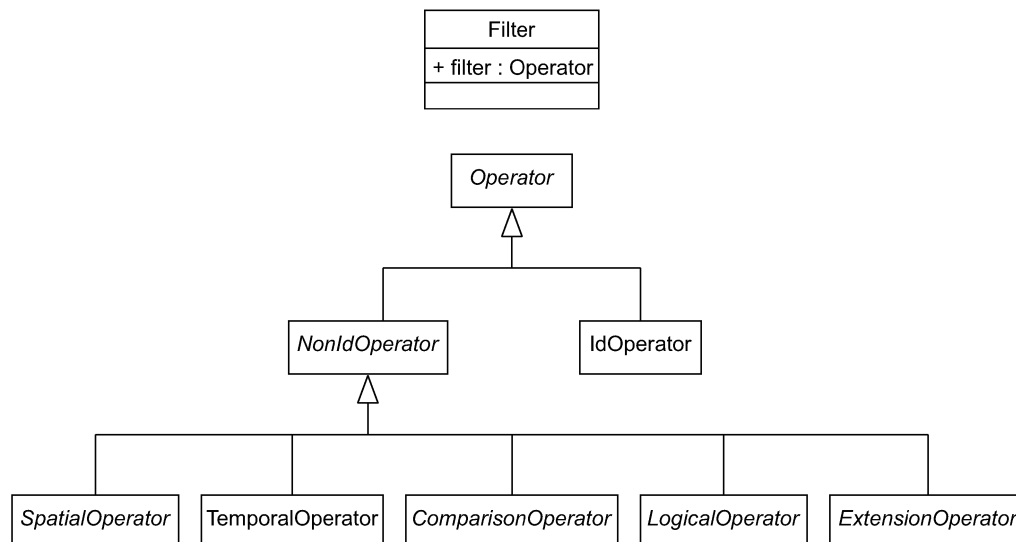


Figure 3 — Filter

7.2 Encoding

The root element of a filter expression, `fes:Filter`, is defined by the following XML Schema fragment:

```

<xsd:element name="Filter"
  type="fes:FilterType"
  substitutionGroup="fes:AbstractSelectionClause"/>
<xsd:complexType name="FilterType">
  <xsd:complexContent>
    <xsd:extension base="fes:AbstractSelectionClauseType">
      <xsd:sequence>
        <xsd:group ref="fes:FilterPredicates"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:group name="FilterPredicates">
  <xsd:choice>
    <xsd:element ref="fes:comparisonOps"/>
    <xsd:element ref="fes:spatialOps"/>
    <xsd:element ref="fes:temporalOps"/>
    <xsd:element ref="fes:logicOps"/>
    <xsd:element ref="fes:extensionOps"/>
    <xsd:element ref="fes:Function"/>
    <xsd:element ref="fes:_Id" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:group>

<xsd:element name="extensionOps"
  type="fes:ExtensionOpsType"
  abstract="true"/>
<xsd:complexType name="ExtensionOpsType" abstract="true"/>

```

The elements contained within the `fes:Filter` element are discussed in detail in subsequent clauses.

7.3 Expressions

7.3.1 General considerations

An expression (see Figure 4) is a combination of one or more symbols that form part of a predicate. In this International Standard, valid symbols shall be encoded using the XML elements defined in this International Standard. Expressions are encoded by nesting these elements to form XML fragments that validate against the schemas in Annex C.

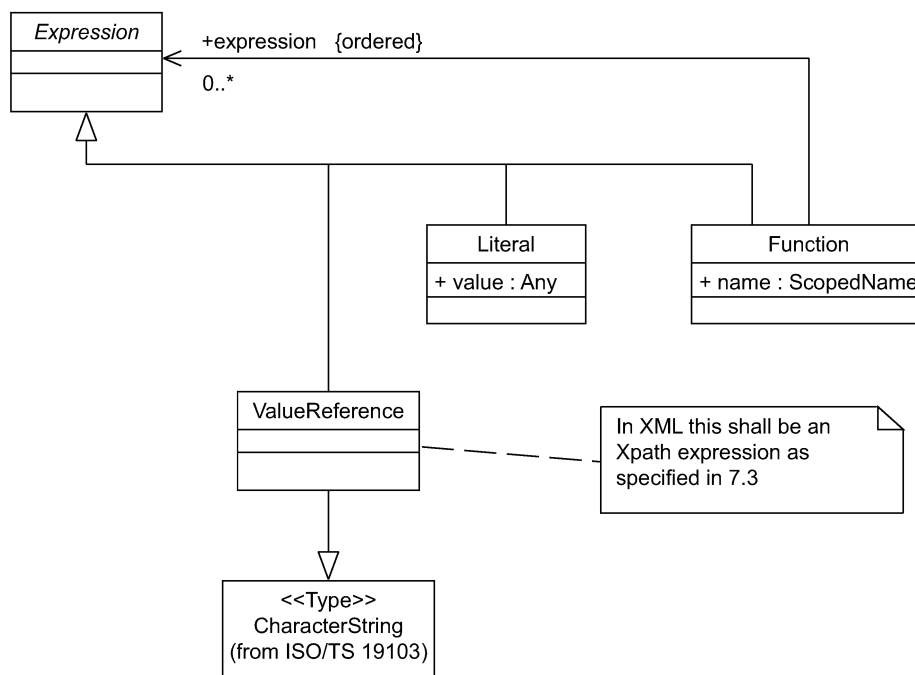


Figure 4 — Expression

7.3.2 Encoding

An expression can be formed using the following XML elements:

- `fes:ValueReference`
- `fes:Literal`
- `fes:Function`

These elements all belong to the substitution group `expression` and can be used wherever an expression is called for. In addition, the XML fragments formed by combining these elements are themselves expressions and can be used wherever an expression is called for.

The **`fes:expression`** element is an abstract element and its only purpose is to act as a placeholder for the elements and combinations of elements that can be used to form expressions.

The XML Schema fragment that defines the abstract `fes:expression` element is:

```
<xsd:element name="expression" abstract="true"/>
```

7.4 Value references

7.4.1 General considerations

A value reference is a string that represents a value that is to be evaluated by a predicate. The string can, for example, be the name of a property of a resource or a path expression that represents some value that is part of the property of a resource. At runtime, a predicate is evaluated by replacing the value reference by the value it refers to and then executing whatever test is encoded by the predicate.

7.4.2 Encoding

The following fragment defines the XML encoding for the `fes:ValueReference` element:

```
<xsd:element name="ValueReference" type="xsd:string" substitutionGroup="fes:expression"/>
```

7.4.3 Property names in GML

The `fes:ValueReference` element can be used to specify the name of any property of an object whose value shall be tested by a predicate in a filter expression. For services that implement this International Standard and use GML (see ISO 19136), property names shall be encoded using XML elements as described in the Extensible Markup Language (XML) 1.0 (see W3C XML) specification. In addition, GML property names may be qualified with a namespace, in which case the name shall conform to the Namespaces in XML (as given in W3C XML Namespaces) specification. The following definition is taken from Clauses 2 and 3 of W3C XML Namespaces:

Names and Tokens

```
[4] NCName ::= (Letter | '_' ) (NCNameChar)*  
/* An XML Name, minus the ":" */  
[5] NCNameChar ::= Letter | Digit | '.' | '-' | '_' | CombiningChar | Extender  
[6] QName ::= (Prefix ':')? LocalPart  
[7] Prefix ::= NCName  
[8] LocalPart ::= NCName
```

The definitions of the components `Letter`, `Digit`, `CombiningChar` and `Extender` are given in Annex B of Namespaces in XML (W3C XML Namespaces).

EXAMPLE 1 Examples of valid property names are: `age`, `temperature`, `_KHz`, `INWATERA_1M.WKB_GEOM`

EXAMPLE 2 Examples of invalid property names are: `+Domain`, `123_SomeName`

7.4.4 XPath expressions

In cases where the data model of the service that implements this International Standard is represented as XML, as is the case with ISO 19142 where GML (see ISO 19136) is used, value references can refer to parts of a complex property and shall be encoded using the XML Path Language (given in W3C XML Path Language).

The XML Path Language (as given in W3C XML Path Language) specification is a language for addressing parts of an XML document, or in the case of this International Standard, for referencing XML elements and attributes that represent the properties of an object encoded in XML.

This International Standard does not require that a filter expression processor support the full XPath language. In order to keep the implementation entry cost as low as possible, services that implement this specification and require the use of XPath, shall support a subset of the XPath language. The following set of rules defines this subset of the XPath language.

- a) The abbreviated form of the child and attribute axis specifier (see W3C XML Path Language) shall be supported.
- b) The context node shall be the resource element, except in the case of a join operation, in which case the context node shall be the parent of the resource element.
- c) Each step in the path may include an XPath predicate.
- d) At least the following predicate expression items shall be supported:
 - 1) a positive non-zero integer may be used to indicate which child of the context node should be selected (i.e. an index). This allows ordered properties with repeatable values to be specifically referenced;
 - 2) an equality predicate for the form “=value” may be used to indicate which child of the context node should be selected based on its value. This allows properties with repeatable value to be specifically referenced by value;