**Open Geospatial Consortium**

Submission Date: <yyyy-mm-dd>

Approval Date:   <yyyy-mm-dd>

Publication Date:   <yyyy-mm-dd>

External identifier of this OGC® document: <<http://www.opengis.net/doc/as/om/3.0>>

Internal reference number of this OGC® document: 20-082r2

Version: 3.0.0

Category: OGC® Abstract Specification

Editors:  Katharina Schleidt**,** Ilkka Rinne

OGC Abstract Specification

Geographic information — Observations, Measurements and Samples

**Copyright notice**

Copyright © 2020 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Document type:    OGC® Abstract Specification

Document subtype:

Document stage:    Draft

Document language:  English

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation. This standard was jointly developed between the OGC and ISO TC 211 and is double branded.

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Submitters

All questions regarding this submission should be directed to the editors or the submitters:

|  |  |
| --- | --- |
| Name | Affiliation |
| Ilkka Rinne (ed.) | Spatineo Oy |
| Katharina Schleidt (ed.) | Datacove |
| Linda van den Brink | Geonovum |
| Sylvain Grellet | BRGM |
| Clemens Portele | Interactive Instruments GmbH |
| Hylke van der Schaaf | Fraunhofer IOSB |

Contributors

The submitters would like to acknowledge the following people as important  
contributors to this version of the specification:

|  |  |
| --- | --- |
| Name | Affiliation |
| Mickael Beaufils | BRGM |
| Hélène Bressan | BRGM |
| Simon Cox | CSIRO Australia |
| Abdelfettah Feliachi | BRGM |
| Robin Huisman | Terraindex B.V. |
| Alistair Ritchie | Landcare Research New Zealand Limited |
| László Sőrés | Hungarian Mining and Geological Service |

Supporting and contributing organizations

The submitters would like to acknowledge the following organizations as important  
supporters and contributors to this version of the specification:

|  |
| --- |
| Name |
| BRGM |
| CSIRO Australia |
| Datacove |
| Finnish Meteorological Institute |
| Fraunhofer IOSB |
| Geonovum |
| Hungarian Mining and Geological Service |
| Interactive Instruments GmbH |
| Manaaki Whenua - Landcare Research New Zealand Limited |
| Spatineo Oy |
| Terraindex B.V. |
| Vaisala Oyj |
| Pole INSIDE - Environmental information systems research center |

**ISO 19156:2020**

ISO TC 211/WG 9

Secretariat: SIS

**Geographic information** — Observations, measurements and samples

CD stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

[Foreword xi](#_Toc86330702)

[Introduction xii](#_Toc86330703)

[1 Scope 1](#_Toc86330704)

[2 Normative references 1](#_Toc86330705)

[3 Terms and definitions 1](#_Toc86330706)

[4 Document conventions 6](#_Toc86330966)

[4.1 Abbreviated terms and acronyms 6](#_Toc86330967)

[4.2 Schema language 6](#_Toc86330968)

[4.3 Model element names 6](#_Toc86330969)

[4.4 Requirements and recommendations 7](#_Toc86330970)

[4.5 Requirements classes 8](#_Toc86330971)

[4.6 Conformance classes 9](#_Toc86330972)

[4.7 Identifiers 9](#_Toc86330973)

[5 Conformance 10](#_Toc86330974)

[5.1 Overview 10](#_Toc86330975)

[5.2 Conformance classes related to models including Observations, Measurements and Samples 10](#_Toc86330976)

[6 Packaging, requirements and dependencies 13](#_Toc86330978)

[6.1 Requirements 13](#_Toc86330979)

[6.2 UML 14](#_Toc86330980)

[6.3 Note on the use of Any 17](#_Toc86330981)

[7 Fundamental characteristics of observations and samples (informative) 18](#_Toc86330982)

[7.1 Observation schema 18](#_Toc86330983)

[7.2 Sample schema 20](#_Toc86330984)

[7.3 Alignment between Observation, Sample and domain models 24](#_Toc86330985)

[8 Conceptual Observation schema 28](#_Toc86330986)

[8.1 General 28](#_Toc86330987)

[8.2 Observation 30](#_Toc86330988)

[8.3 ObservableProperty 36](#_Toc86330989)

[8.4 Procedure 38](#_Toc86330990)

[8.5 ObservingProcedure 38](#_Toc86330991)

[8.6 Observer 40](#_Toc86330992)

[8.7 Host 41](#_Toc86330993)

[8.8 Deployment 43](#_Toc86330994)

[9 Abstract Observation Core 44](#_Toc86330995)

[9.1 General 44](#_Toc86330996)

[9.2 AbstractObservationCharacteristics 45](#_Toc86330997)

[9.3 AbstractObservation 50](#_Toc86330998)

[9.4 AbstractObservableProperty 53](#_Toc86331000)

[9.5 AbstractObservingProcedure 54](#_Toc86331001)

[9.6 AbstractObserver 55](#_Toc86331002)

[9.7 AbstractHost 56](#_Toc86331003)

[9.8 AbstractDeployment 57](#_Toc86331004)

[9.9 NamedValue 58](#_Toc86331005)

[9.10 Codelists 60](#_Toc86331006)

[10 Basic Observations 60](#_Toc86331007)

[10.1 General 60](#_Toc86331008)

[10.2 Observation 62](#_Toc86331009)

[10.3 ObservationCharacteristics 63](#_Toc86331010)

[10.4 ObservationCollection 63](#_Toc86331011)

[10.5 ObservingCapability 65](#_Toc86331012)

[10.6 ObservableProperty 67](#_Toc86331013)

[10.7 ObservingProcedure 68](#_Toc86331014)

[10.8 Observer 69](#_Toc86331015)

[10.9 Host 70](#_Toc86331016)

[10.10 Deployment 71](#_Toc86331019)

[10.11 GenericDomainFeature 72](#_Toc86331022)

[10.12 Codelists 73](#_Toc86331023)

[11 Conceptual Sample schema 77](#_Toc86331024)

[11.1 General 77](#_Toc86331025)

[11.2 Sample 79](#_Toc86331028)

[11.3 Sampling 81](#_Toc86331029)

[11.4 Sampler 84](#_Toc86331030)

[11.5 PreparationStep 85](#_Toc86331031)

[11.6 PreparationProcedure 86](#_Toc86331032)

[11.7 SamplingProcedure 87](#_Toc86331033)

[12 Abstract Sample Core 88](#_Toc86331034)

[12.1 General 88](#_Toc86331035)

[12.2 AbstractSample 89](#_Toc86331038)

[12.3 AbstractSampling 91](#_Toc86331039)

[12.4 AbstractSampler 93](#_Toc86331040)

[12.5 AbstractSamplingProcedure 95](#_Toc86331041)

[12.6 AbstractPreparationProcedure 95](#_Toc86331043)

[12.7 AbstractPreparationStep 96](#_Toc86331046)

[12.8 Codelists 97](#_Toc86331047)

[13 Basic Samples 98](#_Toc86331048)

[13.1 General 98](#_Toc86331049)

[13.2 Sample 99](#_Toc86331052)

[13.3 SpatialSample 100](#_Toc86331053)

[13.4 MaterialSample 102](#_Toc86331054)

[13.5 StatisticalSample 104](#_Toc86331055)

[13.6 Sampling 105](#_Toc86331056)

[13.7 Sampler 106](#_Toc86331057)

[13.8 SamplingProcedure 107](#_Toc86331058)

[13.9 PreparationProcedure 108](#_Toc86331062)

[13.10 PreparationStep 108](#_Toc86331066)

[13.11 SampleCollection 109](#_Toc86331071)

[13.12 PhysicalDimension 110](#_Toc86331072)

[13.13 NamedLocation 111](#_Toc86331073)

[13.14 StatisticalClassification 113](#_Toc86331074)

[13.15 Codelists 114](#_Toc86331075)

[Annex A (normative) Abstract Test Suite 116](#_Toc86331076)

[A.1 Abstract tests for Conceptual Observation schema package 116](#_Toc86331077)

[A.2 Abstract tests for Abstract Observation core package 118](#_Toc86331078)

[A.3 Abstract tests for Basic Observations package 120](#_Toc86331079)

[A.4 Abstract tests for Conceptual Sample schema package 123](#_Toc86331080)

[A.5 Abstract tests for Abstract Sample core package 125](#_Toc86331081)

[A.6 Abstract tests for Basic Samples package 127](#_Toc86331082)

[Annex B (informative) Common usage of OMS concepts 131](#_Toc86331083)

[B.1 Introduction 131](#_Toc86331084)

[B.2 Earth Observations (EO) 131](#_Toc86331085)

[B.3 Metrology 132](#_Toc86331086)

[B.4 Earth science simulations 132](#_Toc86331087)

[B.5 Assay/Chemistry 132](#_Toc86331088)

[B.6 Geology field observations 133](#_Toc86331089)

[B.7 Geotechnics observations 134](#_Toc86331090)

[B.8 Water quality observations 135](#_Toc86331091)

[B.9 Soil quality observations 136](#_Toc86331092)

[Annex C (informative) Changes in the Observation and Sample models between ISO 19156:2011, edition 1 and ISO 19156:2020, edition 2 138](#_Toc86331093)

[C.1 Package and requirements class structure 138](#_Toc86331094)

[C.2 Interfaces in the conceptual schema packages 139](#_Toc86331095)

[C.3 Realizations of the conceptual schemas as abstract and concrete feature type classes 140](#_Toc86331096)

[C.4 Modelling of the Observation concept 140](#_Toc86331097)

[C.5 Modelling of the Sample and Sampling concepts 145](#_Toc86331098)

[C.6 Observation and Sample collections 151](#_Toc86331099)

[C.7 Hard-typing vs. soft typing and codelist use 152](#_Toc86331100)

[C.8 Generic metadata associations 155](#_Toc86331101)

[C.9 Discarded concepts 155](#_Toc86331102)

[Annex D (informative) Best practices in use of the Observation and Sampling models 157](#_Toc86331103)

[D.1 Features, coverages and observations — Different views of information 157](#_Toc86331104)

[D.2 Observation concerns 159](#_Toc86331105)

[D.3 Sample, Sampling concerns 161](#_Toc86331106)

[D.4 Observations and Coverages 163](#_Toc86331107)

[Bibliography 166](#_Toc86331108)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](https://www.iso.org/foreword-supplementary-information.html).

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics****,***  Working Group 9, *Information management*,  in collaboration with the Open Geospatial Consortium, Inc. (OGC).

This second edition cancels and replaces the first edition (ISO 19156:2011), which has been technically revised. The UML model as well as the requirements/conformance class structure has been completely redesigned to address the contemporary modelling and observation data provision use cases. The fundamental Observation model has remained largely the same as in the previous version, with carefully designed improvements and clarifications for the intended use. The Sample model has also been refined. Given the integral nature of the Sample model, is has been decided to include that term in the name of the standard.

Special care has been taken to ensure smooth migration from the ISO 19156:2011 to this version.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html).

Introduction

This International Standard arises from work originally undertaken through the Open Geospatial Consortium’s Sensor Web Enablement (SWE) activity. A set of interfaces and protocols was standardized through which applications and services are able to access sensors of all types, and observations generated by them, over the Web.

A new generation of geospatial standards is now emerging, based on general Web standards, architecture, and current practice, as described in <https://www.w3.org/TR/sdw-bp/>. This includes several new standards for describing and publishing sensors and observations, such as the OGC SensorThings API [12] and the W3C/OGC Semantic Sensor Network Ontology [18]. This new version of the Observations and Measurements Standard (now named “Observations, Measurements and Samples”, OMS for short) is informed by these recent developments. The focus of this revision is aimed at enabling the publication of observation data as part of the Web of data, while also supporting other means of data exchange.

The content presented here derives from the previous version published by Open Geospatial Consortium as OGC 10-004r3, OGC Abstract Specification Geographic information — Observations and measurements (ISO 19156:2011). A technical note describing the changes from the earlier version is available as Annex C.

Geographic information — Observations, Measurements and Samples

# Scope

This International Standard defines a conceptual schema for observations, for features involved in the observation process, and for features involved in sampling when making observations. These provide models for the exchange of information describing observation acts and their results, both within and between different scientific and technical communities.

Observations commonly involve sampling of an ultimate feature-of-interest. This International Standard defines a common set of sample types according to their spatial, material (for ex-situ observations), or statistical nature. The schema includes relationships between sample features (sub-sampling, derived samples).

This International Standard concerns only externally visible interfaces and places no restriction on the underlying implementations other than what is needed to satisfy the interface specifications in the actual situation.

# Normative references

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

ISO 19103:2015, Geographic information — Conceptual schema language

ISO 19107:2019, Geographic information — Spatial schema

ISO 19108:2002, Geographic information — Temporal schema

ISO 19109:2015, Geographic information — Rules for application schema

# Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at <https://www.iso.org/obp>

— IEC Electropedia: available at <http://www.electropedia.org/>

3.1

application schema

conceptual schema for data required by one or more applications

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

3.2

coverage

feature that acts as a function to return values from its range for any direct position within its domain

[SOURCE: ISO/DIS 19123-1, 4.1.9]

3.3

data type

specification of a value domain with operations allowed on values in this domain

EXAMPLE Integer, Real, Boolean, String, Date.

NOTE Data types include primitive predefined types and user-definable types.

[SOURCE: ISO 19103:2015, 4.14]

3.4

domain

well-defined set

Note 1 to entry: Domains are used to define the domain set and range set of attributes, operators and functions.

[SOURCE: ISO 19109:2015, 4.8]

NOTE: All elements within a domain (set) are of a given type.

[SOURCE: ISO/DIS 19123-1, 4.1.16]

3.5

domain feature

feature of a type defined within a particular application domain

NOTE This may be contrasted with observations and sampling features, which are features of types defined for cross-domain purposes.

[SOURCE: ISO 19156:2011, 4.4]

3.6

ex-situ

off-site

referring to the study, maintenance or conservation of a specimen or population away from its natural surroundings.

NOTE Opposite of in-situ (on-site).

[SOURCE: ISO 19156:2011, 4.5, modified — addition of “off-site” and ”on-site”]

3.7

feature

abstraction of real-world phenomena

Note 1 to entry: A feature may occur as a type or an instance. In this document, feature instance is meant unless otherwise specified.

[SOURCE: ISO 19101-1:2014, 4.1.11, modified — Note 1 to entry has been modified.]

3.8

Feature-of-interest

subject of the observation.

3.9

feature type

class of features having common characteristics

[SOURCE: ISO 19156:2011, 4.7]

3.10

measure

<GML> value described using a numeric amount with a scale or using a scalar reference system

Note 1 to entry: When used as a noun, measure is a synonym for physical quantity.

[SOURCE: ISO 19136-1:2020, 3.1.41]

3.11

measurand

quantity intended to be measured

[SOURCE: ISO 19156:2011, 3.11]

3.12

measurement

set of operations having the object of determining the value of a quantity

[SOURCE: ISO 19101-2:2018, 3.21]

3.13

Observation

act carried out by an observer to determine the value of an observable property of an object (feature-of-interest) by using a procedure; the value is provided as the result.

3.14

Observer

identifiable entity that can generate observations pertaining to an observable property by implementing a procedure.

NOTE: An observer is an instance of a sensor, instrument, implementation of an algorithm or a being such as a person.

3.15

Procedure

description of steps performed.

3.16

property

facet or attribute of an object referenced by a name

[SOURCE: ISO 19143:2010, 4.21, modified — Example has been added to the entry.]

EXAMPLE Abby's car has the colour red, where "colour red" is a property of the car.

3.17

property type

characteristic of a feature type

NOTE 1 The value for an instance of an observable property type can be estimated through an act of observation.

NOTE 2 In chemistry-related applications, the term "determinand" or "analyte" is often used.

NOTE 3 Adapted from ISO 19109:2005.

EXAMPLE Cars (a feature type) all have a characteristic colour, where "colour" is a property type.

[SOURCE: Adapted from ISO 19109:2005]

3.18

Proximate feature-of-interest

entity that is directly of interest in the act of observing.

Note 1 to entry: this is a specialized form of the feature-of-interest

3.19

range

〈coverage〉

set of feature attribute values associated by a function, the coverage, with the elements of the domain of a coverage

NOTE This is consistent with the more generic definition of range in 19107 and 19136.

[SOURCE: ISO/DIS 19123-1, 4.1.47]

3.20

Sample

object that is representative of a concept, real-world object or phenomenon.

3.21

Sampler

device or entity (including humans) that is used by, or implements, a sampling procedure to create or transform one or more sample(s).

3.22

sensor

element of a measuring system that is directly affected by a phenomenon, body, or substance carrying a quantity to be measured

[SOURCE: ISO/IEC Guide 99:2007, 3.8, modified — EXAMPLES and NOTE deleted.]

3.23

Ultimate feature-of-interest

entity that is ultimately of interest in the act of observing.

Note 1 to entry: this is a specialized form of the feature-of-interest

3.24

unit of measure

reference quantity chosen from a unit equivalence group

Note to entry: In positioning services, the usual units of measurement are either angular units or linear units. Implementations of positioning services must clearly distinguish between SI units and non-SI units. When non-SI units are employed, it is requird that their relation to SI units be specified.

[ORIGIN: ISO/TC 211 Glossary of Terms - English (last updated: 2020-06-02)]

3.25

value

element of a type domain

[SOURCE: ISO/IEC 19501:2005, 0000\_5, modified — Note 3 has been added to the entry.]]

NOTE 1 A value considers a possible state of an object within a class or type (domain).

NOTE 2 A data value is an instance of a datatype, a value without identity.

NOTE 3 A value can use one of a variety of scales including nominal, ordinal, ratio and interval, spatial and temporal. Primitive datatypes can be combined to form aggregate datatypes with aggregate values, including vectors, tensors and images.

# Document conventions

## Abbreviated terms and acronyms

GFM General Feature Model

GML Geography Markup Language

INSPIRE INfrastructure for SPatial InfoRmation in Europe

O&M Observations and Measurements (ISO 19156:2011)

OMS Observations, Measurements and Samples (this current version of ISO 19156)

OGC Open Geospatial Consortium

SensorML OGC Sensor Model Language

SOS OGC Sensor Observation Service

STA OGC SensorThings API

SWE OGC Sensor Web Enablement

UML Unified Modeling Language

XML Extensible Markup Language

2-D Two Dimensional

3-D Three Dimensional

## Schema language

The conceptual schema specified in this International Standard is in accordance with the Unified Modelling Language (UML) ISO/IEC 19501, following the guidance of ISO 19103:2015.

The UML in Abstract Core and Basic packages is conformant with the profile described in ISO 19136-1:2020, Annex E. Use of this restricted idiom supports direct transformation into a GML Application Schema. ISO 19136 [28] introduces some additional stereotypes. In particular «FeatureType» implies that a class is an instance of the «metaclass» GF\_FeatureType (ISO 19109), and therefore represents a feature type.

The prose explanation of the model uses the term “property” to refer to both class attributes and association roles. This is consistent with the General Feature Model described in ISO 19109. In the context of properties, the term “value” refers to either a literal (for attributes whose type is simple), or to an instance of the class providing the type of the attribute or target of the association. Within the explanation, the property names (property types) are sometimes used as natural language words where this assists in constructing a readable text.

## Model element names

This International Standard specifies a model for observations using terminology that is based on current practice in a variety of scientific and technical disciplines. It is designed to apply across disciplines, so the best or "most neutral" term has been used in naming the classes, attributes and associations provided. The terminology does not, however, correspond precisely with any single discipline. As an aid to implementers, a mapping from the element names specified in this International Standard to common terminology in some application domains is provided in Annex B.

## Requirements and recommendations

All requirements are **normative**, and each is presented with the following template:

|  |  |
| --- | --- |
| **Requirement /req/{pkg}/{classM}/{reqN}** | [Normative statement] |

where **/req/{pkg}/{classM}/{reqN}** identifies the requirement. The use of this layout convention allows the normative provisions of this standard to be easily located by implementers.

All defined classes, attributes and associations mentioned within requirements or recommendations are shown in **bold** correspond to references to the definition of the referenced element.

The following base (/req/{pkg}/) has been used per package:

* **/req/obs-cpt**: Conceptual Observation schema,
* **/req/obs-core**: Abstract Observation Core,
* **/req/obs-basic**: Basic Observations,
* **/req/sam-cpt**: Conceptual Sample schema,
* **/req/sam-core**: Abstract Sample core,
* **/req/sam-basic**: Basic Samples

In the lines below, the base (/req/{pkg}/) has been left out for better readability.

For naming of individual requirements pertaining to classes, the following syntax is used:

* **{Class Name}-sem**: The semantic definition of the concept, together with the naming of the Class.

For naming of individual requirements pertaining to attribute or associations, the following syntax is used:

* **{Attribute/Association Name}-sem**: The semantic definition of the concept, together with the naming of the attribute or association role. Except for cases where concepts are mandatory within all packages, these statements are phrased to be cardinality neutral, e.g. they also apply to cardinality 0..\*;
* **{Attribute/Association Name}-type**: Type information pertaining to the attribute or association when the type is constrained within one model package;
* **{Attribute/Association Name}-card**: Cardinality information pertaining to the attribute or association;
* **{Attribute/Association Name}-con**: Additional constraints. As these sometimes pertain to multiple attributes or associations, this part of the name may become more complex.

Please note that the individual requirements are case sensitive, following UML naming conventions. Requirements pertaining to classes contain the class name in UpperCamelCase, requirements pertaining to associations utilize the association role name in lowerCamelCase.

All recommendations are **informative**, and each is presented with the following template:

|  |  |
| --- | --- |
| **Recommendation /rec/{pkg}/{classM}/{recO}** | [Informative statement] |

where **/rec/{pkg}/{classM}/{recO}** identifies the recommendation. The use of this layout convention allows the informative provisions of this standard to be easily located by implementers.

## Requirements classes

Each statement (requirement or recommendation) in this standard is a member of a requirements class.

All requirement classes are normative.

Each requirements class is described in a discrete clause, and summarized using the following template:

|  |  |
| --- | --- |
| **Requirements class** | **/req/{pkg}/{classM}** |
| Target type | [artefact or technology type] |
| Name | Name of the requirements class |
| Imports | /req/{pkg}/{classZ} |
| Requirement | /req/{pkg}/{classM}/{reqN} |
| Recommendation | /rec/{pkg}/{classM}/{recO} |
| Requirement | /req/{pkg}/{classM}/{reqP} |
| Requirement /Recommendation | [repeat as necessary] |

All requirements in a class must be satisfied. Hence, the requirements class is the unit of re-use and dependency.

Dependency to another requirement class (and the requirements and recommendations defined in it) is done using the “Imports” keyword. All requirements in a dependency SHALL also be satisfied by a conforming implementation.

A requirements class may consist only of dependencies and introduce no new requirements.

## Conformance classes

Conformance to this standard is possible at a number of levels, specified by conformance classes (Annex A). Each conformance class is summarized using the following template:

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/{pkg}/{classM}** |
| Requirements | [identifier for the requirements class] |
| Test purpose | [Reason for test] |
| Test method | [Method to determine if test fulfilled] |
| Test Type | [Type of test] |

All tests in a class must be passed. Each conformance class tests conformance to a set of requirements packaged in a requirements class.

## Identifiers

Each requirements class, requirement and recommendation is identified by a unique identifier. This allows cross-referencing of class membership, dependencies, and links from each conformance test to the requirements tested. Appended to a base URI that identifies the specification as a whole, it enables the construction of a complete URI for identification in an external context.

The entire Requirements and Conformance Structure, consisting of the individual requirements and definitions together with the information on how these are linked together for the creation of Requirements and Conformance classes, will be exposed in a machine actionable format (such as the one provided by the OGC Definitions Server).

The URI for each requirements class has the form:

**http://www.opengis.net/spec/om/3.0/req/{pkg}/{classM}**

The URI for each requirement has the form:

**http://www.opengis.net/spec/om/3.0/req/{pkg}/{classM}/{reqN}**

The URI for each recommendation has the form:

**http://www.opengis.net/spec/om/3.0/rec/{pkg}/{classM}/{recO}**

The URI for each conformance class has the form:

**http://www.opengis.net/spec/om/3.0/conf/{pkg}/{classM}**

# Conformance

## Overview

Clauses 7 to 13 of this International Standard use the Unified Modeling Language (UML) to present conceptual schemas for describing Observations. These schemas define conceptual classes that:

a) may be considered to comprise a cross-domain application schema, or

b) may be used in application schemas, profiles and implementation specifications.

This flexibility is controlled by a set of UML types that can be implemented in a variety of manners. Use of alternative names that are more familiar in a particular application is acceptable, provided that there is a one-to-one mapping to classes and properties in this International Standard.

The UML model in this International Standard defines conceptual classes. Various software systems define implementation classes or data structures. All of these reference the same information content. The same name may be used in implementations as in the model, so that types defined in the UML model may be used directly in application schemas.

Annex A defines a set of conformance tests that will support applications whose requirements range from the minimum necessary to define data structures to full object implementation.

## Conformance classes related to models including Observations, Measurements and Samples

The conformance rules for Models in general are described in ISO 19109:2015. Application Schemas also claiming conformance to this International Standard shall also conform to the rules specified in Clauses 7 to 13 and pass all relevant test cases of the Abstract Test Suite in Annex A.

Depending on the characteristics of the implementing model application, schema or profile, one or more of the declared conformance classes can be chosen for fine-grained Observations, Measurements and Samples conformance. , , , , , and list all of these the classes by package, their relative identifiers and the corresponding subclauses of the Abstract Test Suite. The full URIs of the conformance classes is formed by prefixing the relative URI path as described in Clause 4.7.

**Table 1 — Conceptual Observation schema conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Conceptual Observation schema package | /conf/obs-cpt | A.1.1 |
| Conceptual Observation - Deployment | /conf/obs-cpt/Deployment | A.1.2 |
| Conceptual Observation - Host | /conf/obs-cpt/Host | A.1.3 |
| Conceptual Observation - ObservableProperty | /conf/obs-cpt/ObservableProperty | A.1.4 |
| Conceptual Observation - Observation | /conf/obs-cpt/Observation | A.1.5 |
| Conceptual Observation - Observer | /conf/obs-cpt/Observer | A.1.6 |
| Conceptual Observation - ObservingProcedure | /conf/obs-cpt/ObservingProcedure | A.1.7 |
| Conceptual Observation - Procedure | /conf/obs-cpt/Procedure | A.1.8 |

**Table 2 — Abstract Observation core conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Abstract Observation core package | /conf/obs-core | A.2.1 |
| Abstract Observation core - AbstractDeployment | /conf/obs-core/AbstractDeployment | A.2.2 |
| Abstract Observation core - AbstractHost | /conf/obs-core/AbstractHost | A.2.3 |
| Abstract Observation core - AbstractObservableProperty | /conf/obs-core/AbstractObservableProperty | A.2.4 |
| Abstract Observation core - AbstractObservation | /conf/obs-core/AbstractObservation | A.2.5 |
| Abstract Observation core - AbstractObservationCharacteristics | /conf/obs-core/AbstractObservationCharacteristics | A.2.6 |
| Abstract Observation core - AbstractObserver | /conf/obs-core/AbstractObserver | A.2.7 |
| Abstract Observation core - AbstractObservingProcedure | /conf/obs-core/AbstractObservingProcedure | A.2.8 |
| Abstract Observation core - NamedValue | /conf/obs-core/NamedValue | A.2.9 |

**Table 3 — Basic Observations conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Basic Observations package | /conf/obs-basic | A.3.1 |
| Basic Observations - Deployment | /conf/obs-basic/Deployment | A.3.2 |
| Basic Observations - GenericDomainFeature | /conf/obs-basic/GenericDomainFeature | A.3.3 |
| Basic Observations - Host | /conf/obs-basic/Host | A.3.4 |
| Basic Observations - ObservableProperty | /conf/obs-basic/ObservableProperty | A.3.5 |
| Basic Observations - Observation | /conf/obs-basic/Observation | A.3.6 |
| Basic Observations - ObservationCharacteristics | /conf/obs-basic/ObservationCharacteristics | A.3.7 |
| Basic Observations - ObservationCollection | /conf/obs-basic/ObservationCollection | A.3.8 |
| Basic Observations - Observer | /conf/obs-basic/Observer | A.3.9 |
| Basic Observations - ObservingCapability | /conf/obs-basic/ObservingCapability | A.3.10 |
| Basic Observations - ObservingProcedure | /conf/obs-basic/ObservingProcedure | A.3.11 |

**Table 4 — Conceptual Sample schema conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Conceptual Sample schema package | /conf/sam-cpt | A.4.1 |
| Conceptual Sample - PreparationProcedure | /conf/sam-cpt/PreparationProcedure | A.4.2 |
| Conceptual Sample - PreparationStep | /conf/sam-cpt/PreparationStep | A.4.3 |
| Conceptual Sample - Sample | /conf/sam-cpt/Sample | A.4.4 |
| Conceptual Sample - Sampler | /conf/sam-cpt/Sampler | A.4.5 |
| Conceptual Sample - Sampling | /conf/sam-cpt/Sampling | A.4.6 |
| Conceptual Sample - SamplingProcedure | /conf/sam-cpt/SamplingProcedure | A.4.7 |

**Table 5 – Abstract Sample core conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Abstract Sample core package | /conf/sam-core | A.5.1 |
| Abstract Sample core - AbstractPreparationProcedure | /conf/sam-core/AbstractPreparationProcedure | A.5.2 |
| Abstract Sample core - AbstractPreparationStep | /conf/sam-core/AbstractPreparationStep | A.5.3 |
| Abstract Sample core - AbstractSample | /conf/sam-core/AbstractSample | A.5.4 |
| Abstract Sample core - AbstractSampler | /conf/sam-core/AbstractSampler | A.5.5 |
| Abstract Sample core - AbstractSampling | /conf/sam-core/AbstractSampling | A.5.6 |
| Abstract Sample core - AbstractSamplingProcedure | /conf/sam-core/AbstractSamplingProcedure | A.5.7 |

**Table 6 — Basic Samples conformance classes**

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Identifier** | **Annex A clause** |
| Basic Samples package | /conf/sam-basic | A.6.1 |
| Basic Samples - MaterialSample | /conf/sam-basic/MaterialSample | A.6.2 |
| Basic Samples - NamedLocation | /conf/sam-basic/NamedLocation | A.6.3 |
| Basic Samples - PhysicalDimension | /conf/sam-basic/PhysicalDimension | A.6.4 |
| Basic Samples - Sample | /conf/sam-basic/Sample | A.6.5 |
| Basic Samples - SampleCollection | /conf/sam-basic/SampleCollection | A.6.6 |
| Basic Samples - Sampler | /conf/sam-basic/Sampler | A.6.7 |
| Basic Samples - Sampling | /conf/sam-basic/Sampling | A.6.8 |
| Basic Samples - SpatialSample | /conf/sam-basic/SpatialSample | A.6.9 |
| Basic Samples - StatisticalClassification | /conf/sam-basic/StatisticalClassification | A.6.10 |
| Basic Samples - StatisticalSample | /conf/sam-basic/StatisticalSample | A.6.11 |

# Packaging, requirements and dependencies

## Requirements

As OMS implementations often seamlessly integrate with existing data ecosystems, a very flexible requirements and conformance structure is defined. This structureenables users to selectively mix and match elements as required for their purposes from the OMS data model without the necessity of achieving compliance with the entire data model.

Such flexibility is becoming increasingly relevant with the shift to Linked Data practices, where different organizations maintain and expose only certain aspects of a larger distributed dataset.

EXAMPLE A provider may only serve information on Observable Properties or Monitoring Facilities, while relying on other partners to provide information on measurement procedures. These could claim compliance to those parts falling under their responsibility, while letting other data providers link to these resources.

For this purpose, a fine grained structure for requirements and recommendations, requirements classes and conformance classes has been defined. As far as possible, patterns from the OGC Modular Specification [15] have been taken into account. However, pertaining to the alignment between UML Packages and Conformance Classes, a relaxation of the requirement on one-to-one alignment between UML Package and Conformance Class has been proposed as follows:

* For each UML Package, both a Requirements Class as well as a Conformance Class have been defined;
* Additional Requirements Classes have been created for each Class appearing in the data model, Conformance Classes are added accordingly to enable grouping of the formers and support references to either a group or an individual Requirement Class depending on the need;
* Thematic Domains may create additional Requirements and Conformance Classes reflecting their domain profiles by reference to existing Requirements and Requirements Classes.

As mentioned, as data provision paradigms increasingly shift towards distributed and linked approaches, stipulating that all aspects of an information system conform explicitly to the same underlying standards becomes increasingly difficult. Simultaneously, as distributed data provision becomes increasingly ubiquitous, ever more communities are emerging dedicated to individual aspects of the wider data provision landscape.

One example of such external definition and hosting pertains to the provision of observable properties. In previous versions of the O&M Model, the observable properties concept was only included as a metaclass, with the assumption that a reference to an existing code list will be provided. Within the current OMS Model, the observable property has been upgraded to a featureType. This is because emerging requirements show the need for a more detailed model for this concept. Simultaneously, other communities, such as the Research Data Alliance (RDA), are also working on observable property models (I-ADOPT). The same rationale can be applied to most concepts from the OMS Model.

In order to expose this flexibility beyond the package structure described above, a fine grained hierarchical requirements class structure was created. A modular requirements class is provided for each concept at all three levels of the model. In addition, a further requirements class that imports all the modular classes provided for the individual concepts has been provided for each package.

## UML

### UML package structure

OMS provides the relevant concepts for the structured description of observations, including the sampling structure often essential for true understanding of the nature of the observations being provided. As data provision mechanisms are transitioning towards highly distributed linked approaches, the model structure and packaging has been significantly abstracted. This approach allows implementers to explicitly select the concepts to be supported based on their requirements, while clearly stating to which requirements and Conformance Classes their implementation complies. Both the Observation and Sample sections of this model have been structured using the following layering of packages:

* **Conceptual**: Within the Conceptual Model Packages, only Interfaces are provided. These models provide a very abstract view of the individual concepts they contain without reference to specific implementations. This approach allows for the inclusion of semantically aligned objects from external sources, that while not having been created under the Observations, Measurements and Samples Model, do provide concepts sharing the same semantic meaning as the concepts from the Conceptual Models;
* **Abstract Core**: Within the Abstract Core Model Packages, only abstract featureTypes are provided following the semantic structure of the Conceptual model (i.e: realizing the interfaces provided by the Conceptual Model Packages). A consistent approach to metadata provision is introduced. All associations from the abstract featureTypes reference the conceptual Interfaces for greater implementation flexibility. The Abstract Core Model Packages are foreseen for the creation of domain models providing an Abstract Core ready for Extension;
* **Basic**: Within the Basic Packages, simple concrete featureTypes (specializing the abstract ones from the Abstract Core model) have been defined with some basic utility attributes added for rapid out-of-the-box deployment. A few additional concepts pertaining to collections and potential observations are introduced at this level.

### UML package dependencies

Some model elements used in the schema are defined in other International Standards. The **Table 7** lists the dependencies between the UML packages defined in this International Standard and other International Standards, and the **Figure 1** show the dependencies of the entire OMS UML model package to the other International Standards in a graphical form.

**Table 7 — UML package level dependencies**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dependency from package** | **to package** | **in an International Standard** | **Notes** |
| Conceptual Observation schema | Any type | ISO 19103:2015 (Edition 1) | Any |
| Conceptual Observation schema | Temporal Objects | ISO 19108:2002 (Edition 1) | TM\_Object |
| Conceptual Observation schema | Name types | ISO 19103:2015 (Edition 1) | GenericName |
| Abstract Observation core | Conceptual Observation schema | This International Standard | TM\_Instant, TM\_Period via the Temporal Objects dependency |
| Abstract Observation core | General Feature Model | ISO 19109:2015 (Edition 2) | Feature concepts |
| Abstract Observation core | Text | ISO 19103:2015 (Edition 1) | CharacterString |
| Basic Observations | Abstract Observation core | This International Standard |  |
| Basic Observations | Web environment | ISO 19103:2015 (Edition 1) | URI |
| Basic Observations | Geometry | ISO 19107:2019 (Edition 2) | Geometry |
| Conceptual Sample schema | Any type | ISO 19103:2015 (Edition 1) | Any |
| Conceptual Sample schema | Temporal Objects | ISO 19108:2002 (Edition 1) | TM\_Object |
| Conceptual Sample schema | Name types | ISO 19103:2015 (Edition 1) | GenericName |
| Conceptual Sample schema | Conceptual Observation schema | This International Standard | Observation, Procedure |
| Abstract Sample core | Conceptual Sample schema | This International Standard |  |
| Abstract Sample core | General Feature Model | ISO 19109:2015 (Edition 2) | Feature concepts |
| Abstract Sample core | Geometry | ISO 19107:2019 (Edition 2) | Geometry |
| Abstract Sample core | Text | ISO 19103:2015 (Edition 1) | CharacterString |
| Abstract Sample core | Abstract Observation core | This International Standard | NamedValue |
| Basic Samples | Abstract Sample core | This International Standard |  |
| Basic Samples | Web environment | ISO 19103:2015 (Edition 1) | URI |
| Basic Samples | Measure types | ISO 19103:2015 (Edition 1) | Measure |



**Figure 1 - External UML package dependencies**

## Note on the use of Any

The UML models defined in this International Standard make extensive use of the Any interface defined in ISO 19103:2015 Any type package. The realized Any values of the associations with role names **proximateFeatureOfInterest**, **ultimateFeatureOfInterest**, **result**, **metadata**, **featureOfInterest** and **sampledFeature** MAY be of any type or a reference to an digital representation of an appropriate concept. In the case they are of feature type, the values are not owned by the instances of referring classes, they may have an independent life span from the referring classes, and they may be associated with more than one instance of referring classes.

NOTE any type can be owl:Thing, featureType, dataType

EXAMPLES:

* Reference to SWEET Ontology: <http://sweetontology.net/realmAtmoBoundaryLayer#planetaryboundarylayer>
* Reference to SensorThings deployment: <https://lubw-frost.docker01.ilt-dmz.iosb.fraunhofer.de/v1.1/Locations(269)>
* Reference to 19115 Metadata [24] [25]: <https://inspire-geoportal.ec.europa.eu/resources/INSPIRE-61494ff5-6fad-11e8-b649-52540023a883_20210415-080302/services/1/PullResults/701-750/43.iso19139.xml>
* Reference to an instance of Borehole : <https://data.geoscience.fr/id/borehole/BSS001REWW>
* Reference to an hydro station : <https://iddata.eaufrance.fr/id/HydroStation/Y251002001>
* Reference to a river segment : <https://iddata.eaufrance.fr/id/WatercourseLinkSequence/A0080300>
* An (embedded) Boolean value as Result
* An (embedded) SWE DataRecord
* Elevation Coverage from an external WCS as an observation Result: <https://inspire.rasdaman.org/rasdaman/ows?service=WCS&version=2.0.1&request=GetCoverage&coverageId=INSPIRE_EL&subset=E(494500,496000)&subset=N(4654300,4655000)&format=image/jpeg>
* OMS MaterialSample -> Reference to a rock sample : <https://www.geodata.rocks/Samples/SD-5054_1_A_564_7WR_20-40>

# Fundamental characteristics of observations and samples (informative)

## Observation schema

### Property evaluation

Properties of a feature fall into two basic categories:

1. Value (e.g. name, price, legal boundary) assigned by some authority. These are exact.
2. Value (e.g. height, classification, colour) determined by application of an observation procedure.

These are estimates, with a finite error associated with the value.

The observation error typically has a systematic component, which is similar for all estimates made using the same procedure, and a random component, associated with the particular application instance of the observation procedure. If potential errors in a property value are important in the context of a data analysis or processing application, then the details of the act of observation which provided the estimate of the value are required.

### Observation

An observation is an act associated with a discrete time instant or period through which a number, term or other symbol is assigned to a characteristic. This act involves application of a specified procedure, such as a sensor, instrument, algorithm or process chain. The procedure may be applied in-situ, remotely, or ex-situ with respect to the sampling location. The result of an observation is an estimate of the value of a property of some feature; an observation is a property-value-provider for a feature-of-interest. Use of a common model allows observation data using different procedures to be combined unambiguously.

In conventional measurement theory (e.g. [1][4][5][6][8][13]) the term “measurement” is used. However, a distinction between measurement and category-observation has been adopted in more recent work [2][7][14] so the term “observation” is used here for the general concept. “Measurement” may be reserved for cases where the result is a numerical quantity.

The observation itself is also a feature, since it has properties and identity.

Observation details are important for data discovery and for data quality estimation.

The observation could be considered to carry “property-level” instance metadata, which complements the dataset-level and feature-level metadata commonly provided via catalogue services (e.g. ISO 19115 or other community agreed one).

### Properties of an Observation

An observation results in a value being assigned to a characteristic. The characteristic is a property of a feature, the latter being the feature-of-interest of the observation. The observation uses a specified procedure performed by an observer, which is often an instrument or sensor [1][2], but may be a process chain, human observer, an algorithm, a computation or simulator. The key idea is that the observation result is an estimate of the value of some quality (property, characteristic)of the feature-of-interest, and the other properties of the observation provide context or metadata to support evaluation, interpretation and use of the result. The diagram below (based on representation work done under INSPIRE [19]) provides a visual overview of this.

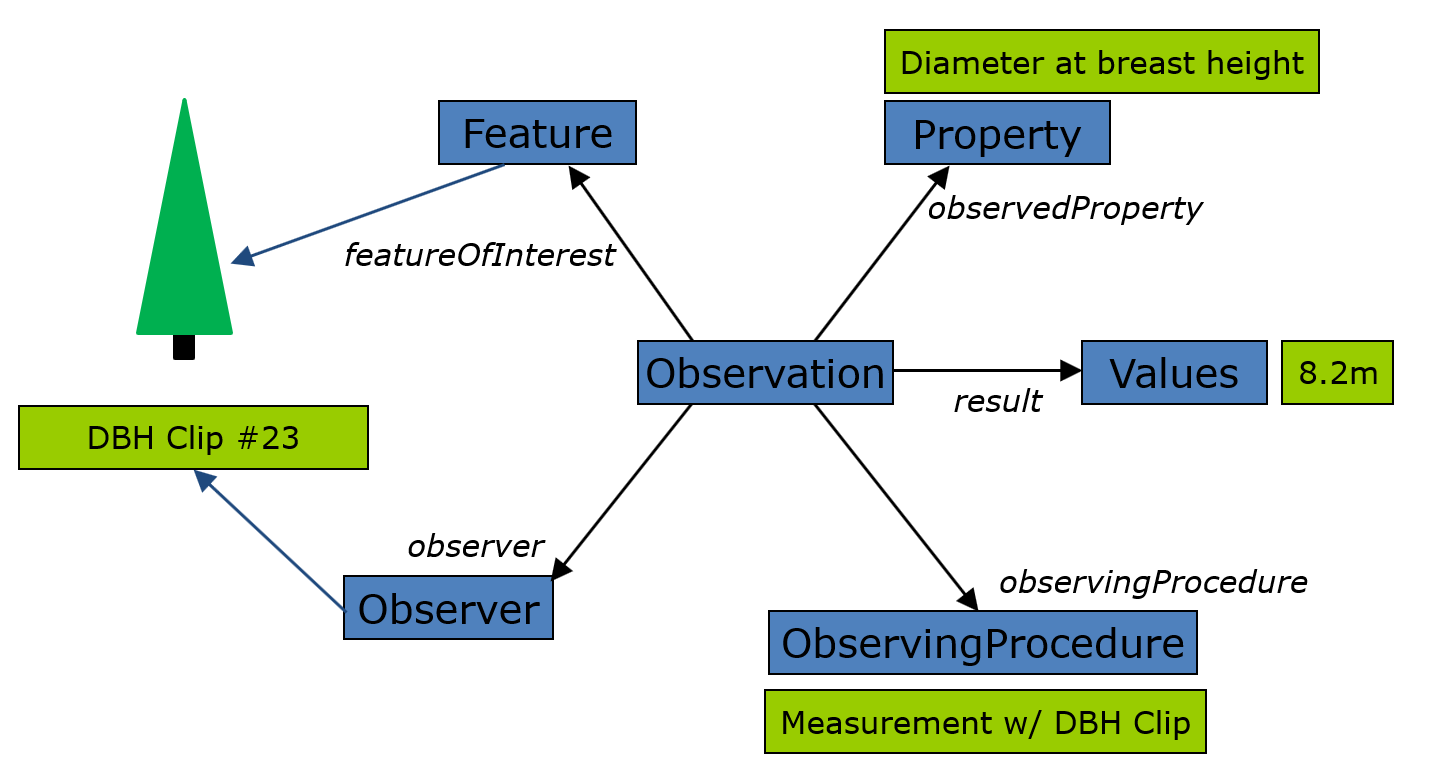


Figure 2 — Properties of an Observation

The relationship between the properties of an observation and those of its feature-of-interest is key to the semantics of the data model elaborated in this standard. This is further elaborated in Annex D.3.

### Observation location

The principal location of interest of an observation is usually associated with the ultimate feature-of-interest.

However, the location of the feature-of-interest may not be readily available. For example, in remote sensing applications, a complex processing chain is required to geolocate the scene or swath. In feature-detection applications the initial observation may be made on a scene, but the entity to be detected, which is the ultimate feature-of-interest, occupies some location within it. The distinction between the proximate and ultimate feature-of-interest is a key consideration in these cases. The proximate feature-of-interest is the object upon which the measurement is directly performed, whereas the ultimate feature-of-interest is the object for which this measurement is ultimately seen as representative of.

Other locations may be relevant in various scenarios. Sub-sampling at locations within the feature-of-interest may occur. The procedure may involve a sensor located remotely from the ultimate feature-of-interest such as in remote sensing, or where specimens are removed from their sampling location and observations made ex-situ (the sampling schema description below elaborates on this). Furthermore, the location of the feature-of-interest may be time-dependent.

The model is generic. The geospatial location of the feature-of-interest may be of little or no interest for some observations (e.g. live specimens, observations made on non-located things like chemical species).

For these reasons, a generic Observation class does not have an inherent location property. Relevant location information should be provided by the feature-of-interest, by the sampling procedure, or by the observation procedure, according to the specific scenario.

NOTE In contrast to spatial properties, some temporal properties are associated directly with an observation (Clauses 8.2.3; 8.2.4). This is due to the fact that an observation is a kind of ‘event’ so its temporal characteristics are fundamental, rather than incidental.

### Result types

Observation results may have many datatypes, including primitive types like category or measure, but also may have more complex types such as time, location and geometry. Complex results are obtained when the observed property requires multiple components for its encoding. Furthermore, if the property varies on the feature-of-interest, then the result is a coverage, whose domain extent is the extent of the feature. In reality, the result will typically be sampled discretely on the domain, and may be represented as a discrete coverage.

Building on this, specialized observation types can be defined by communities to describe the type of result provided, expressed using a terminology common to that community.

### Use of the observation model

The observation model takes a data-user-centric viewpoint, emphasizing the semantics of the feature-of-interest and its properties. This contrasts with sensor-oriented models, which take a process- and thus a provider-centric viewpoint.

The digital representation of an observation is a property-value-provider for a feature-of-interest. Aside from the result, the details of the observation event are primarily of interest in applications where an evaluation of errors in the estimate of the value of a property is of concern. The observation could be considered to carry “property-level” instance metadata, complementing the dataset-level and feature-level metadata that have been conventionally considered (e.g. ISO 19115-1:2014 [24] [25]).

Additional discussion of the application of the observation and sample models, and nuances within these, is provided in Annex D.

## Sample schema

### Role of sample features

A sample may act as a proxy for the ultimate feature-of-interest of an observation, and be associated with this observation by the role feature-of-interest. In this case the sampled-feature association of sample would point upwards in the chain of sampled features leading to ultimate feature-of-interest of the observation. The sample may also be associated with observations, both those being made directly on the sample as well as observations on other samples.

### Proximate vs. ultimate feature-of-interest

#### Introduction

The observation model maps the result of the application of a procedure to a subject, which plays the role of feature-of-interest of the observation. However, the proximate feature-of-interest of an observation may not be the ultimate domain-specific feature whose properties are of interest in the investigation of which the observation is a part. There are three circumstances that can lead to this:

1. The observation does not obtain values for the whole of a domain feature;
2. The observation is performed on a proxy that is not part of the domain feature;
3. The observation procedure obtains values for properties that are not characteristic of the type of the ultimate feature.

Furthermore, in some practical situations, multiple differences apply.

#### Proximate feature-of-interest embodies a sample design

For various reasons, the domain feature may not be fully accessible. In such circumstances, the procedure for estimating the value of a property of the domain feature involves sampling in representative locations. Then the procedure for transforming a property value observed on the sample to an estimate of the property on the ultimate feature-of-interest depends on the sample design.

EXAMPLE 1 The chemistry of water in an underground aquifer is sampled at one or more positions in a well or bore.

EXAMPLE 2 The magnetic field of the earth is sampled at positions along a flight-line. In contrast to the well in the example above, the flight-line does not represent a real-world object.

EXAMPLE 3 The structure of a rock mass is observed on a cross-section exposed in a river bank.

EXAMPLE 4 The bubble of air around the intake of an air quality monitoring station is taken as representative for the wider air around the station. Again, a virtual feature serves as proximate feature-of-interest.

In other cases, where direct observation of the domain feature is not possible, the observation may be performed on a proxy.

EXAMPLE 5 In order to measure the intensity of the sun’s light, the reflectance on a white sheet of paper may be utilized as a proxy for the sun’s intensity.

In some cases, the observation procedure obtains values for properties that are not characteristic of the type of the ultimate feature

EXAMPLE 6 The salinity of water in a well is measured, the feature-of-interest of this well is an aquifer. However, the final target of the observation is the fluid body contained within the aquifer (see **Figure 8**).

#### Observed property is a proxy

The procedure for obtaining values of the property of interest may be indirect, relying on direct observation of a more convenient parameter which is a proxy for the property of interest. Application of an algorithm or processing chain obtains an estimate of the ultimate property of interest.

The observation model requires that the feature-of-interest of the initial observation be of a type that carries the observed property within its properties. Thus, if the proxy property is not a member of the ultimate feature-of-interest, a proxy feature with a suitable model shall be involved.

EXAMPLE 1 A remote sensing observation might obtain the reflectance colour, when the investigation is actually interested in vegetation type and quality. The feature which contains reflectance colour is a scene or swath, while the feature carrying vegetation properties is a parcel or tract.

EXAMPLE 2 The direct value coming from a sensor may be quantified as a voltage, whereas the observed property represented by this voltage is the physiochemical value being observed by the sensor (ex : pH).

#### Combination

These variations may be combined if exhaustive observation of the domain feature is impractical, and direct measurement is of a proxy property.

EXAMPLE For certain styles of mineralization, the gold concentration of rocks in a region might be estimated through measurement of a related element (e.g. copper), in a specimen of gravel collected from a stream that drains part of the region. The gravel samples the rocks in the catchment of the stream, i.e. in the stream bed and upslope.

### Role of Sample

Samples are artefacts of an observational strategy, and have no significant function outside of their role in the observation process. The physical characteristics of the samples themselves are of little interest, except perhaps to the manager of a sampling campaign.

EXAMPLE 1 In various countries/domains, terms like “site” and station” are encountered. These usually correspond to an identifiable locality where a monitoring facility (host, platform, ...) has been established, sensors, or other measurement devices (observers) have been deployed to acquire observations on a given observable property applying a specific procedure. In the context of the observation model, the spatial sample (both proximate and ultimate) connotes the “world in the vicinity of the observer/sampler”, so the observed properties relate to the physical medium at the observer/sampler described by the sample, and not to any physical artefact such as a mooring, buoy, benchmark, monument, well, and so forth that may be described by Host.

EXAMPLE 2 In some domains, elements are taken from their natural environment (ex-situ) curated and preserved for the purpose of keeping a trace of their existence. Examples are biodiversity studies, crop seed preservation, and so forth. In those cases, the material samples considered are called specimen. That’s why the class named SF\_Specimen in the previous version of the standard is renamed into MaterialSample in this updated version.

EXAMPLE 3 Statistical samples usually apply to populations or other sets, of which certain subset may be of specific interest.

NOTE A transient spatial sample, such as a ships-track or flight-line, might be identified and described, but is unlikely to be revisited exactly.

A sample is intended to sample some object in an application domain. However, in some cases the identity, and even the exact type, of the sampled object may not be known when observations are made using the sample.

### Sampling process

Understanding the process by which samples are obtained is often essential to understanding the context of subsequent measurements on this object (feature-of-interest). Different sampling strategies can provide vastly different samples, in turn leading to different result values in observations pertaining to these samples.

A sample is created through the act of sampling, whereby a sampler follows a defined procedure in order to identify and/or extract representative samples from the ultimate feature-of-interest.

The nature of the sampler varies by sampling strategy; at one end of the spectrum the sampler can be a sensor or other automated measurement device; at the other end of the spectrum the sampler can be a human being providing observations or taking part in a biodiversity survey campaign.

As a dependence on the sampling strategy, a sampling procedure appropriate to the sampling act to be performed must be selected and defined. For the provision of fine-grained information pertaining to the sampling process, multiple sampling procedures can be applied to one sampling act. Multiple sampling procedures may also be required for the case where one sampling process classifies samples in accordance with multiple criteria.

EXAMPLE When performing observations on populations, these may first be sampled by gender and age. Sampling Procedures describing the criteria utilized for gender and age classification can be provided individually.

A sampling event may involve very different samples, whereby some of these samples may serve purely to provide contextual information pertaining to the sampling event.

EXAMPLE When sampling water from a river, information on the meteorology at the time of sampling may be relevant for the interpretation of measurements obtained on the water sample.

### Classification of samples

A small number of common sampling patterns, similar across domains, provide a basis for processing and portrayal tools, and depend particularly on the geometry of the sample design. These provide a basis for processing and portrayal tools which are similar across domains, and depend particularly on the geometry of the sample design. Common names for sampling features include specimen, sample, site, profile, transect, path, swath and scene.

Spatial sampling is classified primarily by the topological dimension. Material samples may provide information on their original source location, but are more often characterized by their size and storage location.

In addition, various preparation steps may be performed on samples both before and after observations are performed on the sample.

Additional information on provenance, curation and methods of archiving a sample has been delegated to external standards that may be referenced via the ‘metadata’ association that can be provided for all types contained within the Sampling model.

## Alignment between Observation, Sample and domain models

### Model consistency

The type of the feature-of-interest is defined in an application schema (ISO 19109). This may be part of a domain model, or may be from a cross-domain model, such as Sample (Clause 11). The feature type defines its set of characteristics as properties. For consistency, the feature-of-interest shall carry the observed property as part of the definition of its type (e.g. **Figure 4**).

EXAMPLE

A pallet with the characteristic mass is to be described via a feature model. In the simplest form, an interface “Pallet” may be defined as having the attribute “mass” of type “Measure” describing the mass characteristic of the pallet being described (**Figure 3**). However, when using this direct approach, no further measurement metadata is available, only the numeric mass is provided together with the unit of measurement.



**Figure 3 — (Example) Pallet interface, simple example for model consistency.**

Alternatively, through utilization of the OMS model, an observation providing the value of this property for the feature being investigated may be utilized to fulfil the data requirements ensuing from the Pallet Interface. This approach makes it possible for the information system to ‘describe’ how the result (here mass value) was obtained together with the relevant value.

For this purpose, the observation shall have observedProperty “mass”, the result shall be of the type “Measure” and the scale (unit of measure) shall be suitable for mass measurements. Thus, the requirements ensuing from the Pallet Interface are fulfilled, while additional relevant measurement meta-information is also provided; model consistency has been ensured. This approach is illustrated in **Figure 4**.



**Figure 4 — (Example) An observation with consistent properties: the observed property (mass) is a characteristic associated with the type of the feature-of-interest (Pallet) and the procedure and result type are also suitable.**

The figure below shows a complete representation of a mass observation. In addition to the basic information provided with the observation in the preceding diagram, information on the specific measurement device used is provided together with information on where this device was deployed as the observation was performed.



**Figure 5 — (Example) An observation with complete properties: for additional context, the Observer, Host and Deployment have been added**

An attribute from within the conceptual model can be instantiated as an Observation in the concrete realization. The attributes that have been defined for the domain feature within the interface, in the example “mass” and “uom”, can be realized through the association of an observation carrying this information. Formally, these two representations both realize the defined interface.

Based on the use case, when modelling one must decide whether solely providing information of type ‘Measure’ with uom is sufficient for the domain considered. In some cases, the full OMS model is required to actually discover, exchange and reuse data properly. For example, a single attribute ‘lake surface’ will be sufficient for most mapping agency needs whereas a more thorough observation description of how that surface was measured and when (e.g.: dam empty/full, rainfall observation…) is important for water management needs.

### Relationship between Sample and domain features

A sample feature is established in order to make observations concerning some domain feature. The association “sampledFeature” links the sample feature to the feature which the sampling feature was designed to sample. The target of this association is usually a real-world feature from an application domain (see **Figure 6**).

EXAMPLE A profile typically samples a water- or atmospheric-column; a well samples the water in an aquifer; a tissue specimen samples a part of an organism.



**Figure 6 — (Informative) Relationship between Sample and domain features.**

Both the Sample and the Domain feature may appear as the feature-of-interest. If a Sample feature is involved, it samples a feature of a type defined in a domain model.

Any Domain object can be a featureOfInterest of an Observation.

The more refined example described in **Figure 7** further explains how both Sample and Observation from the OMS model can interact with a domain model.

In this example, Well, Aquifer and FluidBody are modelled outside the OMS model but:

* The Well also conforms to the Sample requirements;
* Instances from the domain model are the proximate and ultimate features of interest of the WaterSalinity Observation.

The Well that samples the Aquifer acts as a proxy to the Aquifer in the observation act. The Well is thus considered as the proximateFeatureOfInterest of the Observation. The sampledFeature (the Aquifer) of the Well being the ultimateFeatureOfInterest.



**Figure 7 — (Example) Sampling Cascade example including domain features.**

Depending on the use case, one might want to push the modelling choice a step further and instantiate a FluidBody in the system according to the semantic of the domain model (Well, Aquifer, FluidBody). That example is further refined in **Figure 8**. Then depending on the viewpoint considered, either the instance of the Aquifer and/or the instance of the FluidBody can be considered as the ultimateFeatureOfInterest of the Observation. The Well remains the proximateFeatureOfInterest.



**Figure 8 — (Example) Complex Sampling Cascade example referencing external domain feature.**

# Conceptual Observation schema

## General

### Conceptual Observation model

The Conceptual Observation schema is described as a class diagram in **Figure 9**. The schema is fully described in the Clause 8.1.2.



**Figure 9 — Conceptual Observation schema overview.**

### Conceptual Observation schema package Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt |
| Target type | Conceptual model |
| Name | Conceptual Observation schema package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Observation |
| Imports | /req/obs-cpt/ObservableProperty |
| Imports | /req/obs-cpt/Procedure |
| Imports | /req/obs-cpt/ObservingProcedure |
| Imports | /req/obs-cpt/Observer |
| Imports | /req/obs-cpt/Host |
| Imports | /req/obs-cpt/Deployment |

### Association relatedObservation

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/gen/relatedObservation-sem | An **Observation** the object is related to.  If a reference to a related **Observation** is provided, the association with role **relatedObservation** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

## Observation

### Observation Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/Observation |
| Target type | Conceptual model |
| Name | Conceptual Observation - Observation |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Requirement | /req/obs-cpt/Observation/Observation-sem |
| Requirement | /req/obs-cpt/Observation/phenomenonTime-sem |
| Requirement | /req/obs-cpt/Observation/phenomenonTime-card |
| Requirement | /req/obs-cpt/Observation/resultTime-sem |
| Requirement | /req/obs-cpt/Observation/resultTime-card |
| Requirement | /req/obs-cpt/Observation/validTime-sem |
| Requirement | /req/obs-cpt/Observation/featureOfInterest-sem |
| Requirement | /req/obs-cpt/Observation/featureOfInterest-card |
| Requirement | /req/obs-cpt/Observation/observedProperty-sem |
| Requirement | /req/obs-cpt/Observation/observedProperty-card |
| Requirement | /req/obs-cpt/Observation/result-sem |
| Requirement | /req/obs-cpt/Observation/result-card |
| Requirement | /req/obs-cpt/Observation/observingProcedure-sem |
| Requirement | /req/obs-cpt/Observation/observingProcedure-card |
| Requirement | /req/obs-cpt/Observation/observer-sem |
| Requirement | /req/obs-cpt/Observation/host-sem |
| Recommendation | /rec/obs-cpt/Observation/observerhost-con |
| Recommendation | /rec/obs-cpt/Observation/observedProperty-con |
| Recommendation | /rec/obs-cpt/Observation/observingProcedure-con |
| Recommendation | /rec/obs-cpt/Observation/result-con |
| Recommendation | /rec/obs-cpt/Observation/phenomenonTimeResult-con |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |
| Requirement | /req/obs-cpt/Observation/uom |
| Recommendation | /rec/obs-cpt/Observation/uom-con |

### Interface Observation

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/Observation-sem | An observation is an act carried out by an **Observer** to determine the value of an **ObservableProperty** of an object (**featureOfInterest)** by using an **ObservingProcedure**; the value is provided as the **result**. |

NOTE It is important to note that the terms ‘observation’, ‘interpretation’, ‘forecast’, ‘simulation’ do correspond to this definition. This aspect is further clarified in Clause 7 “Fundamental characteristics of observations and samples (informative)”.

### Attribute phenomenonTime

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/phenomenonTime-sem | The time that the **result** applies to the characteristic of the **FeatureOfInterest** being observed.  If the **phenomenonTime** is described, this SHALL be provided by the attribute **phenomenonTime:TM\_Object** |

NOTE 1 The phenomenonTime is often the time the Sample has been taken.

NOTE 2 This is often the time of interaction by a sampling Procedure or observation Procedure with a real-world feature.

NOTE 3 If the result is the average of multiple samples taken at different times, then the phenomenonTime is the time interval over which these measurements were taken.

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/phenomenonTime-card | An **Observation** SHALL have exactly 1 **phenomenonTime.** |

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/phenomenonTimeResult-con | If the **observedProperty** of an **Observation** is ‘occurrence time’ then the **result** SHOULD be the same as the **phenomenonTime.** |

### Attribute resultTime

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/resultTime-sem | The instant of time when the **result** of the **Observation** became available.  If the **resultTime** is described, this SHALL be provided by the attribute **resultTime:TM\_Object** |

EXAMPLE 1 The resultTime typically corresponds to when the Procedure associated with the Observation was completed. For some observations this is identical to the phenomenonTime. However, there are important cases where they differ.

EXAMPLE 2 Where a measurement is made on a specimen in a laboratory, the phenomenonTime is the time the specimen was retrieved from its host, while the resultTime is the time the laboratory procedure was applied.

EXAMPLE 3 The resultTime also supports disambiguation of repeat measurements made of the same property of a feature using the same procedure.

EXAMPLE 4 Where sensor observation results are post-processed, the resultTime is the post-processing time, while the phenomenonTime is the time of initial interaction with the world.

EXAMPLE 5 Simulations may be used to estimate the values for phenomena in the future or past. The phenomenonTime is the time that the result applies to, while the resultTime is the time that the simulation was executed.

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/resultTime-card | An **Observation** SHALL have exactly 1 **resultTime**. |

### Attribute validTime

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/validTime-sem | The time interval during which the **result** is assumed to be applicable for use.  If **validTime(s)** are described they SHALL be provided by the attribute **validTime:TM\_Period** |

NOTE This attribute is commonly required in forecasting applications.

### Association featureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/featureOfInterest-sem | The subject of the **Observation**.  The reference(s) to **featureOfInterest(s)** SHALL be provided using the association **Domain** with the role **featureOfInterest**. |

NOTE 1 The featureOfInterest can be of Any type

EXAMPLE 1 An instance of a feature modelled in a specific domain model (Borehole according to OGC GeoSciML).

EXAMPLE 2 The bubble of air around the intake of an air quality monitoring station.

EXAMPLE 3 An existing well being used for water quality measurements.

NOTE 2 This object is the real-world object whose properties are under observation, or is an object created with the intention to sample the real-world object, as described in the Sampling part of section 7. An observation instance serves as a propertyValueProvider for its feature-of-interest.

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/featureOfInterest-card | An **Observation** SHALL have at least 1 **featureOfInterest** and MAY have more than 1 in cases objects are created with the intention to sample the real-world object  The cardinality of the **featureOfInterest** association SHALL be 1 at minimum. |

### Association observedProperty

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/observedProperty-sem | The **ObservableProperty** that is the subject of the **Observation**.  If a reference to an **ObservableProperty** is provided, the association with the role **observedProperty** SHALL be used. |

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/observedProperty-card | An **Observation** SHALL have exactly 1 **observedProperty**. |

### Association result

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/result-sem | The **result** of the **Observation**.  If a reference to a **result** is provided, the association **Range** with the role **result** SHALL be used. |

NOTE 1 The result can be of Any type as it may represent the value of any feature property.

NOTE 2 If the observed property is a spatial operation or function, the type of the result may be a coverage.

NOTE 3 In some contexts, particularly in earth and environmental sciences, the term “observation” is used to refer to the result itself.

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/result-card | An **Observation** SHALL have exactly 1 **result** . |

### Association observingProcedure

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/observingProcedure-sem | The **ObservingProcedure** used by the **Observation** to determine the value of the **ObservableProperty** provided by the **result.**  If a reference to an **ObservingProcedure** is provided, the association with the role **procedure** SHALL be used. |

EXAMPLE Observed radiance wavelength is determined by the response characteristics of the sensor.

A description of the observation procedure provides or implies an indication of the reliability or quality of the observation result.

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/observingProcedure-card | An **Observation** SHALL have exactly 1 **observingProcedure**. |

### Association observer

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/observer-sem | An **Observer (ref)** that is involved in the creation of this **Observation**.  If a reference to an **Observer** is provided, the association with the role **observer** SHALL be used. |

### Association host

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/host-sem | A **Host** that is involved in the creation of this **Observation**  If a reference to a **Host** is provided, the association with the role **host** SHALL be used. |

### Constraint Observer or Host

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/observerhost-con | At least one **Observer** or **Host** SHOULD be provided |

### Constraint ObservableProperty characteristic associated with featureOfInterest

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/observedProperty-con | The **ObservableProperty** referenced by **observedProperty** SHOULD correspond to a characteristic associated with the **featureOfInterest** |

### Constraint suitable ObservableProperty

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/observingProcedure-con | The **ObservingProcedure** referenced by **procedure** SHOULD be suitable for the associated **ObservableProperty** |

### Constraint suitable result type

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/result-con | The type of the result provided by the **result** association SHOULD be suitable for the associated **ObservableProperty** |

### Constraint unit of measure

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observation/uom | The **Observation** SHALL provide a unit of measure (UoM) if the result is measurable. If the UoM is not contained in the result, it SHALL be provided in the context of the **Observation**; the provision modality is to be defined by communities. |

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-cpt/Observation/uom-con | The unit of measure SHOULD be suitable for the associated **ObservableProperty** and **ObservingProcedure** |

NOTE: in the case where the result of the Observation is a classification, for which no unit exists, the UoM should be declared as unitless (e.g., referencing QUDT [17] <http://qudt.org/vocab/unit/UNITLESS> or UCUM [9] for “no units”).

## ObservableProperty

### ObservableProperty Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/ObservableProperty |
| Target type | Conceptual model |
| Name | Conceptual Observation - ObservableProperty |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-cpt/ObservableProperty/ObservableProperty-sem |
| Requirement | /req/obs-cpt/ObservableProperty/observer-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |

### Interface ObservableProperty

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/ObservableProperty/ObservableProperty-sem | A quality (property, characteristic) of the **feature-of-interest** that can be observed. |

EXAMPLE 1 The height of a tree, the depth of a water body, or the temperature of a surface are examples of observable properties, while the value of a classic car is not (directly) observable but asserted.

EXAMPLE 2 Groundwater Level

On a groundwater well we:

* Monitor Groundwater Level (1 observable property)
* With an automated probe (that remains in the ground all year, constituting 1 procedure).

Then we have physical campaigns where we revisit the groundwater well and:

* Measure the Groundwater Level (still the same observable property as above)  
  but
* With a manual probe, this is a different procedure.

This allows for checking whether the probe needs recalibration.

### Association observer

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/ObservableProperty/observer-sem | An **Observer** capable of observing this **ObservableProperty**.  If a reference to the **Observer** is provided, the association with the role **observer** SHALL be used. |

## Procedure

### Procedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/Procedure |
| Target type | Conceptual model |
| Name | Conceptual Observation - Procedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-cpt/Procedure/Procedure-sem |

### Interface Procedure

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Procedure/Procedure-sem | A description of steps performed. |

NOTES:

* Procedure is an abstract concept that is then further specialized in the various procedure types defined in this document. All share the commonality of describing a defined series of steps to a specific purpose.
* The term process that was used in ISO 19156:2011 was purposely dropped in this version to avoid unnecessary confusion between the terms procedure and process.

## ObservingProcedure

### ObservingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/ObservingProcedure |
| Target type | Conceptual model |
| Name | Conceptual Observation - ObservingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Procedure |
| Requirement | /req/obs-cpt/ObservingProcedure/ObservingProcedure-sem |
| Requirement | /req/obs-cpt/ObservingProcedure/observer-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |

### Interface ObservingProcedure

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/ObservingProcedure/ObservingProcedure-sem | The description of steps performed in order to determine the value of an **observableProperty** by an **Observer**. |

NOTES:

* Depending on the complexity of the use case, the procedure will be more or less explicitly described. Especially pertaining to historical data, there may be very little or no information available - this information should also be provided.
* The recipe that the observer (cook) follows to generate the observation.
* The procedure is often referred to as the method.
* Different observers can follow the same (reusable) procedure for the creation of different observations.
* The procedure is a workflow, protocol, plan, algorithm, or computational method specifying how to make an observation.
* The observing procedure cannot describe a sensor instance, but it can describe the sensor type.

NOTE The term process that was used in ISO 19156:2011 has been purposely dropped in this version to avoid unnecessary confusion between the terms procedure and process.

EXAMPLE An instance of Procedure is a description of the process utilized by an observer, this could be a chemical analysis method, a protocol for measuring an object, but could also be a checklist utilized by a human observer during a biodiversity campaign. Procedure could further describe the algorithms behind simulators or models used to generate a result from other inputs.

### Association observer

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/ObservingProcedure/observer-sem | An **Observer** capable of performing this **ObservingProcedure**.  If a reference to an **Observer** is provided, the association with the role **observer** SHALL be used. |

## Observer

### Observer Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/Observer |
| Target type | Conceptual model |
| Name | Conceptual Observation - Observer |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-cpt/Observer/Observer-sem |
| Requirement | /req/obs-cpt/Observer/observableProperty-sem |
| Requirement | /req/obs-cpt/Observer/observingProcedure-sem |
| Requirement | /req/obs-cpt/Observer/deployment-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |

### Interface Observer

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observer/Observer-sem | An identifiable entity that can generate **Observations** pertaining to an **observableProperty** by implementing an **ObservingProcedure**. |

NOTES

* Different Observers can follow the same (reusable) observing Procedure for the creation of different Observations.
* The Observer is the entity instance, not the entity type. Pertaining to sensors, the Observer would reference the explicit sensor, while the Procedure would reference the methodology utilized by that sensor type.
* An Observer is closely linked with an observableProperty that it generates results for.
* An Observer can be hosted by one or more Host.
* The Observer is an instance of a sensor, instrument, implementation of an algorithm, or a being such as a person.

An Observer responds to a stimulus, e.g., a change in the environment, or input data composed from the results of prior Observations, and generates a result.

EXAMPLE Accelerometers, gyroscopes, barometers, magnetometers, and so forth are Observers that are typically mounted on a modern smartphone (which acts as Host). Other examples of Sensors include the human eyes.

### Association observableProperty

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observer/observableProperty-sem | An **ObservableProperty** that this **Observer** can observe.  If a reference to **ObservableProperty**(s)is provided, the association with the role **observableProperty** SHALL be used. |

### Association observingProcedure

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observer/observingProcedure-sem | An **ObservingProcedure** that this **Observer** can perform.  If a reference to **ObservingProcedure**(s)is provided, the association with the role **observingProcedure** SHALL be used. |

### Association deployment

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Observer/deployment-sem | A **Deployment** to which this **Observer** is either physically or organizationally attached.  If a reference to **Deployment**(s)is provided, the association with the role **deployment** SHALL be used. |

## Host

### Host Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/Host |
| Target type | Conceptual model |
| Name | Conceptual Observation - Host |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-cpt/Host/Host-sem |
| Requirement | /req/obs-cpt/Host/deployment-sem |
| Requirement | /req/obs-cpt/Host/relatedHost-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |

### Interface Host

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Host/Host-sem | A grouping of **Observer**s for a specific reason. |

NOTES:

* In many use cases, the Host is the environmental monitoring facility.
* The Host can be a platform that hosts a set of sensors.
* An alternative usage could pertain to a biodiversity survey campaign. In this scenario, the team performing the survey would be modelled as observers whereas the entire survey campaign can be represented as a Host.

### Association deployment

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Host/deployment-sem | A **Deployment** at this **Host**.  If a reference to a **Deployment** is provided, the association with the role **host** SHALL be used. |

### Association relatedHost

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Host/relatedHost-sem | A **Host** the **Host** is related to.  If a reference to a related **Host** is provided, the association with role **relatedHost** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

## Deployment

### Deployment Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-cpt/Deployment |
| Target type | Conceptual model |
| Name | Conceptual Observation - Deployment |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-cpt/Deployment/Deployment-sem |
| Requirement | /req/obs-cpt/Deployment/observer-sem |
| Requirement | /req/obs-cpt/Deployment/host-sem |

### Interface Deployment

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Deployment/Deployment-sem | Information on the assignment of an **Observer** to a **Host.** |

NOTE Examples of deployment are

* Information regarding a sensor being attached to a pole.
* The monitoring facilities pertaining to an environmental monitoring network.
* The description of a ship cruise linking a research vessel with a marine network.
* The participation of a citizen in a citizen-science project involving crowd sensing.

### Association observer

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Deployment/observer-sem | The **Observer** associated with this **Deployment**.  If a reference to an **Observer** is provided, the association with the role **observer** SHALL be used. |

### Association host

|  |  |
| --- | --- |
| **Requirement** /req/obs-cpt/Deployment/host-sem | The **Host** to which this **Deployment** pertains.  If a reference to a **Host** is provided, the association with the role **host** SHALL be used |

# Abstract Observation Core

## General

### Abstract Observation Core Package Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core |
| Target type | Logical model |
| Name | Abstract Observation core package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-core/AbstractObservationCharacteristics |
| Imports | /req/obs-core/AbstractObservation |
| Imports | /req/obs-core/AbstractObservableProperty |
| Imports | /req/obs-core/AbstractObservingProcedure |
| Imports | /req/obs-core/AbstractObserver |
| Imports | /req/obs-core/AbstractHost |
| Imports | /req/obs-core/AbstractDeployment |

### Association metadata

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/gen/metadata-sem | If descriptive metadata is provided, the association role **metadata** SHALL link to descriptive metadata as commonly understood by communities. |

NOTE Attention should be given not to reinvent semantic that is explicitly modelled in the OMS model.

## AbstractObservationCharacteristics

### AbstractObservationCharacteristics Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractObservationCharacteristics |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractObservationCharacteristics |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/AbstractObservationCharacteristics-sem |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/type-sem |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/parameter-sem |
| Recommendation | /rec/obs-core/AbstractObservationCharacteristics/parameter-procedure |
| Recommendation | /rec/obs-core/AbstractObservationCharacteristics/parameter-redundant |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/resultQuality-sem |
| Requirement | /req/obs-cpt/Observation/phenomenonTime-sem |
| Requirement | /req/obs-cpt/Observation/resultTime-sem |
| Requirement | /req/obs-cpt/Observation/validTime-sem |
| Requirement | /req/obs-cpt/Observation/featureOfInterest-sem |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/pFoI-sem |
| Requirement | /req/obs-core/AbstractObservationCharacteristics/uFoI-sem |
| Requirement | /req/obs-cpt/Observation/observedProperty-sem |
| Requirement | /req/obs-cpt/Observation/result-sem |
| Requirement | /req/obs-cpt/Observation/observingProcedure-sem |
| Requirement | /req/obs-cpt/Observation/observer-sem |
| Requirement | /req/obs-cpt/Observation/host-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |
| Requirement | /req/obs-core/gen/metadata-sem |
| Recommendation | /rec/obs-core/AbstractObservationCharacteristics/uFoI |
| Imports | /req/obs-core/NamedValue |



**Figure 10 — Context diagram for Abstract Observation core — AbstractObservationCharacteristics and AbstractObservation.**

### Feature type AbstractObservationCharacteristics

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/AbstractObservationCharacteristics-sem | Set of common characteristics used for describing an **Observation** or a collection of Observations. |

### Attribute observationType

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/type-sem | Information providing further detail on the type of Observations being described by the **AbstractObservationCharacteristics**.  If information on the type of Observation is provided, the property **observationType:AbstractObservationType** SHALL be used. |

NOTE Observation type allows describing the formalism, encoding, etc. to be expected when accessing objects associated to the Observation.

NOTE Multiple types may be applied to one Observation, such as in the case where the Observation is being typed both by the Domain (feature-of-interest geometry) as well as Range (result type).

### Attribute parameter

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/parameter-sem | Arbitrary event-specific parameter relevant to the **AbstractObservationCharacteristics**.  If additional parameter information is provided, the property **parameter:NamedValue** SHALL be used. |

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-core/AbstractObservationCharacteristics/parameter-procedure | Parameter SHOULD NOT be used instead of the procedure to describe the steps performed in order to determine the value of the ObservableProperty. |

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-core/AbstractObservationCharacteristics/parameter-redundant | Parameter SHOULD NOT be utilized to provide information already contained in the model by existing attributes or associations. |

NOTE This might be an environmental parameter, an instrument setting or input, or an event-specific sampling parameter that is not tightly bound to either the feature-of-interest or to the observation procedure.

NOTE Parameters that are tightly bound to the procedure can be recorded as part of the procedure description.

The AbstractObservingProcedure is a generic or standard procedure, rather than an event-specific process. In this context, parameters bound to the observation act, such as instrument settings, calibrations or inputs, local position, detection limits, asset identifier, operator, may augment the description of a standard procedure.

EXAMPLE A time sequence of observations of water quality in a well might be made at variable depths within the well. While these can be associated with specimens taken from the well at this depth as the features-of-interest, a more common approach is to identify the well itself as the feature-of-interest, and add a “samplingDepth” parameter to the observation. The sampling depth is of secondary interest compared to the temporal variation of water quality at the site.

### Attribute resultQuality

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/resultQuality-sem | Information pertaining to the data quality of the **result**.  If additional data quality information is provided, the property **resultQuality:Any** SHALL be used. |

NOTE This instance-specific description complements the description of the observation Procedure, which provides information concerning the quality of all observations using this procedure. The quality of a result may be assessed following the procedures in ISO 19157 [29] [30] [31]. Multiple measures may be provided.

### Association proximateFeatureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/pFoI-sem | The entity that is directly of interest in the act of observing.  If a reference to the entity being directly observed is provided, the association with the role **proximateFeatureOfInterest** SHALL be used.  This association is a specialization of the **featureOfInterest** role. |

NOTE The measurement process may be performed on an intermediary entity referred to as proximateFeatureOfInterest that acts as a proxy to the ultimate feature-of-interest that is being observed (measured, estimated or calculated).

### Association ultimateFeatureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationCharacteristics/uFoI-sem | The entity that is ultimately of interest in the act of observing.  If a reference to the entity ultimately being observed is provided, the association with the role **ultimateFeatureOfInterest** SHALL be used.  This association is a specialization of the **featureOfInterest** role. |

NOTE The measurement process may be performed on an intermediary entity that acts as a proxy to the ultimate feature-of-interest that is being observed (measured, estimated or calculated).

If in the real world both ultimateFeatureOfInterest and proximateFeatureOfInterest exist but not both have a digital representation, then the appropriate relation should be selected that best describes the nature of the entity being referenced.

|  |  |
| --- | --- |
| **Recommendation** /rec/obs-core/AbstractObservationCharacteristics/uFoI | In the case where ultimate and proximate features-of-interest are the same object, the association SHOULD be provided using the **ultimateFeatureOfInterest** association role. |

NOTE There will often be a specifiable relationship between the proximate and ultimate feature-of-interest, such as a sampling-chain, see Clause 7.2.2 for examples.

EXAMPLE A river, an aquifer, soil layer, outcrop, a butterfly, a survey area, a room, Abby's car, a specific human being, this document

EXAMPLE 1: To determine the concentrations of chemical compounds in a river, a sample is taken in a predefined location in the river. This sample is taken to a laboratory where the required chemical analysis is done. In this case, the river is the ultimateFeatureOfInterest, while the sample is the proximateFeatureOfInterest .

EXAMPLE 2: Pertaining to document and observations on the consistency thereof, for the Observation “This clause is inconsistent”, the ultimateFeatureOfInterest is the entire document, while the proximateFeatureOfInterest is the specific clause being addressed.

EXAMPLE 3: the determination of the species of the butterfly, in this case the butterfly is the ultimateFeatureOfInterest, no proximateFeatureOfInterest need be provided.

## AbstractObservation

### AbstractObservation Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractObservation |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractObservation |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Imports | /req/obs-core/AbstractObservationCharacteristics |
| Requirement | /req/obs-cpt/Observation/phenomenonTime-card |
| Requirement | /req/obs-cpt/Observation/resultTime-card |
| Requirement | /req/obs-cpt/Observation/observedProperty-card |
| Requirement | /req/obs-cpt/Observation/observingProcedure-card |
| Requirement | /req/obs-cpt/Observation/result-card |
| Requirement | /req/obs-cpt/Observation/Observation-sem |
| Requirement | /req/obs-core/AbstractObservation/observationType-sem |
| Requirement | /req/obs-core/AbstractObservationType/AbstractObservationType-sem |
| Requirement | /req/obs-core/AbstractObservation/resultTime-type |
| Requirement | /req/obs-core/AbstractObservation/featureOfInterest-con |
| Requirement | /req/obs-core/AbstractObservation/parameterName-card |
| Requirement | /req/obs-core/Observation/observerhost-con |
| Requirement | /req/obs-core/Observation/observedProperty-con |
| Requirement | /req/obs-core/Observation/observingProcedure-con |
| Requirement | /req/obs-core/Observation/result-con |
| Requirement | /req/obs-cpt/Observation/uom |
| Recommendation | /rec/obs-cpt/Observation/uom-con |
| Recommendation | /rec/obs-cpt/Observation/observedProperty-con |
| Recommendation | /rec/obs-cpt/Observation/observerhost-con |
| Recommendation | /rec/obs-cpt/Observation/observingProcedure-con |
| Recommendation | /rec/obs-cpt/Observation/result-con |
| Recommendation | /rec/obs-cpt/Observation/phenomenonTimeResult-con |

### Constraint observationType

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservation/observationType-sem | If information on the type of Observation is provided, the constraints defined in the referenced codelist SHALL be used. |

### Constraint resultTime instant

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservation/resultTime-type | If the result time of the **Observation** is provided, the **resultTime** attribute SHALL be of type **TM\_Instant**. |



### Constraint parameter unique name

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservation/parameterName-card | The **name** attribute of a **parameter NamedValue** SHALL be unique within an **Observation**. |

### Constraint proximate or ultimate featureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservation/featureOfInterest-con | at least one **proximateFeatureOfInterest** or **ultimateFeatureOfInterest** SHALL be given. |

### Constraint Observer or Host

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/Observation/observerhost-con | At least one **Observer** or **Host** SHALL be provided |

### Constraint ObservableProperty characteristic associated with featureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/Observation/observedProperty-con | The **ObservableProperty** referenced by **observedProperty** SHALL correspond to a characteristic associated with the **featureOfInterest** |

### Constraint suitable ObservableProperty

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/Observation/observingProcedure-con | The **ObservingProcedure** referenced by **procedure** SHALL be suitable for the associated **ObservableProperty** |

### Constraint suitable result type

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/Observation/result-con | The type of the result provided by the **result** association SHALL be suitable for the associated **ObservableProperty** |

## AbstractObservableProperty

### AbstractObservableProperty Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractObservableProperty |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractObservableProperty |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/ObservableProperty |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 11 — Context diagram for Abstract Observation core — AbstractObservableProperty.**

## AbstractObservingProcedure

### AbstractObservingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractObservingProcedure |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractObservingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/ObservingProcedure |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 12 — Context diagram for Abstract Observation core — AbstractObservingProcedure.**

## AbstractObserver

### AbstractObserver Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractObserver |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractObserver |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Observer |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 13 — Context diagram for Abstract Observation core — AbstractObserver, AbstractHost and AbstractDeployment.**

## AbstractHost

### AbstractHost Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractHost |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractHost |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Host |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 14 — Context diagram for Abstract Observation core — AbstractHost.**

## AbstractDeployment

### AbstractDeployment Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/AbstractDeployment |
| Target type | Logical model |
| Name | Abstract Observation core - AbstractDeployment |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Imports | /req/obs-cpt/Deployment |
| Requirement | /req/obs-core/AbstractDeployment/deploymentReason-sem |
| Requirement | /req/obs-core/AbstractDeployment/deploymentTime-sem |
| Requirement | /req/obs-core/gen/metadata-sem |

### Attribute deploymentReason

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractDeployment/deploymentReason-sem | A human readable description of the reason for the **Deployment**.  If the reason for the **Deployment** is provided, the property ***deploymentReason:CharacterString***SHALL be used. |

EXAMPLES

* A researcher involved in a biodiversity survey campaign assessing the distribution of selected alien species. The deploymentReason describes the fact that this individual was involved in this campaign for the reason of identifying alien species.
* A sensor is mounted on a building to monitor seismic activities.
* A new sensor type is rolled out within a regional or thematic network due to new legal reporting requirements.

### Attribute deploymentTime

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractDeployment/deploymentTime-sem | The time that the **Deployment** pertains to.  If the time of the **Deployment** is provided, property ***deploymentTime:TM\_Period***SHALL be used. |

EXAMPLES

* A researcher involved in a biodiversity survey campaign assessing the distribution of selected alien species. The deploymentTime provides the time period(s) during which this person carried out this activity in the framework of the campaign.
* A sensor is mounted on a building to monitor seismic activities. The deploymentTime provides the time period(s) during which this sensor is mounted or active.

## NamedValue

### NamedValue Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-core/NamedValue |
| Target type | Logical model |
| Name | Abstract Observation core - NamedValue |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Requirement | /req/obs-core/NamedValue/NamedValue-sem |
| Requirement | /req/obs-core/NamedValue/name-sem |
| Requirement | /req/obs-core/NamedValue/value-sem |

### Data type NamedValue

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/NamedValue/NamedValue-sem | The class **NamedValue** provides for a generic soft-typed parameter value. |

### Attribute name

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/NamedValue/name-sem | The attribute **name:GenericName** SHALL indicate the meaning of the named value. |

NOTE The value of the name should be taken from a well-governed source if possible.

EXAMPLE When used as the value of an Observation:parameter, the name might take values like ‘procedureOperator’, ‘detectionLimit’, ‘amplifierGain’, ‘samplingDepth’, 'analysisIteration', ...

### Attribute value

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/NamedValue/value-sem | The attribute **value:Any** SHALL provide the value. |

NOTE The type “Any” should be substituted by a suitable concrete type, such as CI\_ResponsibleParty or Measure.

## Codelists

### AbstractObservationType

The code list AbstractObservationType can be specialized as required to firm up semantics of observation types, as done in the derived codelist ObservationTypeByResultType below.

|  |  |
| --- | --- |
| **Requirement** /req/obs-core/AbstractObservationType/AbstractObservationType-sem | An empty extension point for providing various classification schemes for **Observations**.  If **Observation** classification schemes are used in the implementing application schemas, a concrete realization SHALL be created for the application. |

# Basic Observations

## General

### Basic Observations Package Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic |
| Target type | Logical model |
| Name | Basic Observations package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-basic/Observation |
| Imports | /req/obs-basic/ObservationCharacteristics |
| Imports | /req/obs-basic/ObservationCollection |
| Imports | /req/obs-basic/ObservingCapability |
| Imports | /req/obs-basic/ObservableProperty |
| Imports | /req/obs-basic/ObservingProcedure |
| Imports | /req/obs-basic/Observer |
| Imports | /req/obs-basic/Host |
| Imports | /req/obs-basic/Deployment |
| Imports | /req/obs-basic/GenericDomainFeature |
| Requirement | /req/obs-basic/ObservationCollectionType/ObservationCollectionType-sem |
| Requirement | /req/obs-basic/ObservationTypeByResultType/ObservationTypeByResultType-sem |
| Requirement | /req/obs-basic/ObservationTypeByResultType/ObservationTypeByResultType-con |
| Requirement | /req/obs-basic/ObservationCollectionType/ObservationCollectionType-sem |
| Requirement | /req/obs-basic/ObservationCollectionType/homogeneous-con |
| Requirement | /req/obs-basic/ObservationCollectionType/summarizing-con |

### Attribute link

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/gen/link-sem | Additional descriptive resources pertaining to a feature.  If a link to a descriptive resource is provided, the attribute **link:URI** SHALL be used. |

### Attribute location

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/gen/location-sem | Location information pertaining to a feature**.**  If location information is provided, the attribute **location:Geometry** SHALL be used. |

## Observation

### Observation Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/Observation |
| Target type | Logical model |
| Name | Basic Observations - Observation |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-core/AbstractObservation |



**Figure 15 — Context diagram for Basic Observations — Observation.**

## ObservationCharacteristics

### ObservationCharacteristics Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/ObservationCharacteristics |
| Target type | Logical model |
| Name | Basic Observations - ObservationCharacteristics |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-core/AbstractObservationCharacteristics |
| Requirement | /req/obs-basic/ObservationCharacteristics/collection-sem |

### Association collection

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCharacteristics/collection-sem | An **ObservationCollection** that is described by these **ObservationCharacteristics**.  If a reference to a collection **ObservationCharacteristics** is provided, the association with the role **collection** SHALL be used. |

## ObservationCollection

### ObservationCollection Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/ObservationCollection |
| Target type | Logical model |
| Name | Basic Observations - ObservationCollection |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/obs-basic/ObservationCollection/ObservationCollection-sem |
| Requirement | /req/obs-basic/ObservationCollection/collectionType-sem |
| Requirement | /req/obs-basic/ObservationCollection/collectionType-con |
| Requirement | /req/obs-basic/ObservationCollection/member-sem |
| Requirement | /req/obs-basic/ObservationCollection/memberCharacteristics-sem |
| Requirement | /req/obs-basic/ObservationCollection/relatedCollection-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |
| Requirement | /req/obs-basic/AbstractObservationCollectionType/AbstractObservationCollectionType-sem |

### Feature type ObservationCollection

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/ObservationCollection-sem | A collection of similar **Observation**s |

### Attribute collectionType

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/collectionType-sem | Information on the type of the **ObservationCollection.**  If information on the collection type is provided, the attribute **collectionType:AbstractObservationCollectionType** SHALL be used. |

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/collectionType-con | If the **collectionType** is provided, property values of the associated **Observation** and **ObservationCharacteristics** instances SHALL comply with the constraints defined for this **collectionType** value. |

### Association member

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/member-sem | An **Observation** that is part of this **ObservationCollection**.  If a reference to a member **Observation** is provided, the association with the role **member** SHALL be used. |

### Association memberCharacteristics

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/memberCharacteristics-sem | Information on **ObservationCharacteristics** of **Observations** contained within the **ObservationCollection**.  If a reference to **ObservationCharacteristics** pertaining to the collection members is provided, the association with the role **memberCharacteristics** SHALL be used. |

### Association relatedCollection

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollection/relatedCollection-sem | A **ObservationCollection** the **ObservationCollection** is related to.  If a reference to a related **ObservationCollection** is provided, the association with role **relatedCollection** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

## ObservingCapability

### ObservingCapability Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/ObservingCapability |
| Target type | Logical model |
| Name | Basic Observations - ObservingCapability |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-basic/ObservationCharacteristics |
| Requirement | /req/obs-basic/ObservingCapability/ObservingCapability-sem |



**Figure 16 — Context diagram for Basic Observations — ObservingCapability and ObservationCollection.**

### Feature type ObservingCapability

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservingCapability/ObservingCapability-sem | Information on **Observation**(s)that could potentially be provided. |

EXAMPLE In order to explicitly describe the capabilities of an Environmental Monitoring Facility, one must provide information on what Observable Properties are being measured with which methodology.

For example in a national groundwater quantity monitoring network, depending on the equipment and the underlying observational strategies:

* Some monitoring may have just one ObservingCapability:
  + ObservingCapability:
    - ultimateFeatureOfInterest: ’Hydrogeological Unit 121AS’,
    - proximateFeatureOfInterest:’xyz’,
    - procedure: ‘Groundwater depth measurement by electronic probe’,
    - observedProperty: ‘GroundWaterDepth’
* Some other, may have several such as
  + ObservingCapability 1:
    - ultimateFeatureOfInterest: ‘Entite hydrogeologique 143AE05’,
    - proximateFeatureOfInterest: ‘Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE’,
    - procedure: ‘Groundwater depth measurement by electronic probe’,
    - observedProperty: ‘GroundWaterDepth’
  + ObservingCapability 2:
    - ultimateFeatureOfInterest: ‘Entite hydrogeologique 143AE05’,
    - proximateFeatureOfInterest: ‘Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE’,
    - procedure: ‘Digital recording teletransmitted’,
    - observedProperty: ‘Water Temperature’
  + ObservingCapability 3:
    - ultimateFeatureOfInterest: ‘Entite hydrogeologique 143AE05’,
    - proximateFeatureOfInterest: ‘Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE’,
    - procedure: ‘Digital recording teletransmitted’,
    - observedProperty: ‘Water conductivity measured at 25°C’

NOTE In the example above, URIs have been removed and only the labels provided for better readability.

## ObservableProperty

### ObservableProperty Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/ObservableProperty |
| Target type | Logical model |
| Name | Basic Observations - ObservableProperty |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Imports | /req/obs-core/AbstractObservableProperty |
| Requirement | /req/obs-basic/gen/link-sem |



**Figure 17 — Context diagram for the Basic Observations — ObservableProperty.**

## ObservingProcedure

### ObservingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/ObservingProcedure |
| Target type | Logical model |
| Name | Basic Observations - ObservingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Imports | /req/obs-core/AbstractObservingProcedure |
| Requirement | /req/obs-basic/gen/link-sem |



**Figure 18 — Context diagram for Basic Observations — ObservingProcedure.**

## Observer

### Observer Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/Observer |
| Target type | Logical model |
| Name | Basic Observations - Observer |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Imports | /req/obs-core/AbstractObserver |
| Requirement | /req/obs-basic/gen/link-sem |
| Requirement | /req/obs-basic/gen/location-sem |



**Figure 19 — Context diagram for Basic Observations — Observer, Host and Deployment.**

## Host

### Host Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/Host |
| Target type | Logical model |
| Name | Basic Observations - Host |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Imports | /req/obs-core/AbstractHost |
| Requirement | /req/obs-basic/gen/link-sem |
| Requirement | /req/obs-basic/gen/location-sem |



Figure 20 – Context diagram for Basic Observations – Host

## Deployment

### Deployment Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/Deployment |
| Target type | Logical model |
| Name | Basic Observations - Deployment |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Imports | /req/obs-core/AbstractDeployment |
| Requirement | /req/obs-basic/gen/link-sem |



Figure 21 – Context diagram for Basic Observations – Deployment

## GenericDomainFeature

### GenericDomainFeature Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/obs-basic/GenericDomainFeature |
| Target type | Logical model |
| Name | Basic Observations - GenericDomainFeature |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Requirement | /req/obs-basic/GenericDomainFeature/GenericDomainFeature-sem |
| Requirement | /req/obs-basic/gen/link-sem |
| Requirement | /req/obs-basic/gen/location-sem |



**Figure 22 — Context diagram for Basic Observations — GenericDomainFeature.**

### Feature type GenericDomainFeature

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/GenericDomainFeature/GenericDomainFeature-sem | A concrete featureType to be utilized as **featureOfInterest** of an **Observation.** |

NOTE This type is foreseen as a placeholder for specialized domain features in order to enable rapid prototyping.

## Codelists

### AbstractObservationCollectionType

The code list AbstractObservationCollectionType can be specialized as required to firm up semantics of collection types, as done in the derived codelist ObservationCollectionType below.

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/AbstractObservationCollectionType/AbstractObservationCollectionType-sem | An empty extension point for providing various classification schemes for **ObservationCollections**.  If **ObservationCollection** classification schemes are used in the implementing application schemas, a concrete realization SHALL be created for the application. |

### ObservationCollectionType

The code list ObservationCollectionType realizes the AbstractObservationCollectionType and has the following values defined in this International Standard: "homogeneous" and "summarizing".

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollectionType/ObservationCollectionType-sem | The following entries SHALL be provided:   * homogeneous: all observations contained are of a similar nature * summarizing: a wider grab-bag type of collection |

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationCollectionType/homogeneous-con | If **collectionType** in the **ObservationCollection** is specified as “homogeneous” from this Codelist, the following constraints apply to the associated **ObservationCharacteristics** and all **Observation** instances referenced via the member association.  If a property value is provided within the **ObservationCharacteristics**, this value applies to all **Observations** contained in the **ObservationCollection**:   * property not provided - values may be provided by the observations but is not provided at this level * property provided but with no content - no observation within the collection provides this property * property = value - this value applies to all observations within the collection * property = value set/range - this value set/range applies to all observations within the collection |

NOTE: the observations need not contain attributes or associations supplied via the ObservationCharacteristics when collectionType is set to homogeneous.

EXAMPLE 1 If the collection has the value “A” for property “foo” then all Observations in the collection have value “A” for that property.

EXAMPLE 2 If the collection states the ObservableProperty X, then all observations contained shall refer to that ObservableProperty.

|  |  |
| --- | --- |
| **Requirement**  /req/obs-basic/ObservationCollectionType/summarizing-con | If **collectionType** in the **ObservationCollection** is specified as “summarizing” from this Codelist, the following constraints apply to the associated **ObservationCharacteristics** and all **Observation** instances referenced via the member association.  If multiple values for a property are available in the contained **Observations**, ALL values for this attribute (or the range of values contained in all **Observations**) are provided in the **ObservationCharacteristics**. A property may also be empty in the **ObservationCharacteristics** - in this case any value can be provided for this attribute within the contained Observations:   * property not provided - values may be provided by the observations but a summary is not provided at this level * property provided but with no content - no observation within the collection provides this property * property = value - this value applies to all observations within the collection * property = value set/range - all observations provide a value within this set/range |

NOTE If a summarizing collection provides a set/range for an attribute it may be that all observations have this exact set/range as value for this attribute, or they could have different values that fall in the set/range.

EXAMPLE 1

If the summarizing collection supplies: phenomenonTime=2020-01-01T00:00:00Z/2020-02-01T00:00:00Z, validTime=[empty/NIL/null] and no other properties, this would mean that:

* Observations in the collection can have any value for the resultTime property, since it is absent from the collection.
* None of the Observations in the collection provide a value for validTime  
  Note: [empty/NIL/null] is a placeholder for the encoding specific representation of the absence of information.
* Observations can have any value for the phenomenonTime property that falls completely in the given time range. Valid examples would be:
  + 2020-01-05T00:00:00+05:00
  + 2020-01-05T10:00:00Z/2020-01-05T11:00:00Z
  + 2020-01-01T00:00:00Z/2020-02-01T00:00:00Z

EXAMPLE 2

If the summarizing collection supplies: result=1, this would mean that all the Observations in the collection have a value of 1 for the result property.

EXAMPLE 3

If the summarizing collection supplies: result=1, 2, 5, [8 - 11] (the values 1, 2 and 5, and the range 8-11), then examples of possible values for the result property on the contained Observations are:

* 1
* 9
* 2, 5 (a set with the two values)
* [8.1 - 9.2] (a range of 8.1 to 9.2)
* 1, 2, 5, [8 - 11] (the exact set of values from the collection)

EXAMPLE 4

If the summarizing collection supplies:

* ultimateFeatureOfInterest=https://example.org/collections/42/items/42,
* deployment=[empty/NIL/null] (i.e. property provided but with no content),
* observer=[https://example.org/v1.1/Sensors/41, https://example.org/v1.1/Sensors/43]

then this means:

* The Observations in the collection all have the same ultimateFeatureOfInterest (a reference to https://example.org/collections/42/items/42).
* None of the Observations in the collection have a (reference to a) deployment.
* All Observations in the collection have either one, or both, of the referenced Observers.
* Since the proximateFeatureOfInterest is not specified in the collection, the Observations in the collection can have any value for this field.

### ObservationTypeByResultType

The code list ObservationTypeByResultType is a specialization of AbstractObservationType created to support the legacy observation types from the previous version of this standard.

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationTypeByResultType/ObservationTypeByResultType-sem | The following entries SHALL be provided:   * measurement: the result is of type Measure. * category-observation: the result is of type ScopedName. * truth-observation: result is a truth value. * count-observation: the result is of type Integer. * temporal-observation: the result is of type TM\_Object. * geometry-observation: the result is of type Geometry * complex-observation: the result is of type Record. * discrete-coverage-observation: result is a coverage that returns the same feature attribute values for every direct position within any single spatial object, temporal object, or spatiotemporal object in its domain. * discrete-point-coverage: result is a coverage that has a domain composed of points, * timeseries-observation: the result is a timeseries (a sequence of data values which are ordered in time). |

|  |  |
| --- | --- |
| **Requirement** /req/obs-basic/ObservationTypeByResultType/ObservationTypeByResultType-con | The following constraints shall be applied to the value of the result association of the Observation based on the codelist value used:   * If the value "measurement" is used, the value of the result shall be of type Measure. * If the value "category-observation" is used the value of the result shall be of type ScopedName. * If the value "truth-observation" is used, the value of result shall be a truth value. * If the value "count-observation" is used, the value of the result shall be of type Integer. * If the value "temporal-observation" is used, the value of the result shall be of type TM\_Object. * If the value "geometry-observation" is used, the value of the result shall be of type Geometry * If the value "complex-observation" is used, the value of the result shall be of type Record. |

# Conceptual Sample schema

## General

### Conceptual Sample schema model

The Conceptual Sample schema described as a class diagram in **Figure 23**. It is fully described in the Clause 11.1.2.



**Figure 23 – Conceptual Sample schema overview.**

### Conceptual Sample Schema Package Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt |
| Target type | Conceptual model |
| Name | Conceptual Sample schema package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-cpt/Sample |
| Imports | /req/sam-cpt/Sampling |
| Imports | /req/sam-cpt/Sampler |
| Imports | /req/sam-cpt/PreparationStep |
| Imports | /req/sam-cpt/PreparationProcedure |
| Imports | /req/sam-cpt/SamplingProcedure |

## Sample

### Sample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/Sample |
| Target type | Conceptual model |
| Name | Conceptual Sample - Sample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Requirement | /req/sam-cpt/Sample/Sample-sem |
| Requirement | /req/sam-cpt/Sample/sampling-sem |
| Requirement | /req/sam-cpt/Sample/preparationStep-sem |
| Requirement | /req/sam-cpt/Sample/sampledFeature-sem |
| Requirement | /req/sam-cpt/Sample/relatedSample-sem |
| Requirement | /req/obs-cpt/gen/relatedObservation-sem |

### Interface Sample

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sample/Sample-sem | A **Sample** is anobject that is representative of a concept, real-world object or phenomenon. |

NOTE

* The way the sample is taken is typically guided by a sampling strategy. Samples are often artefacts of an observational strategy, and often have no significant function outside of their role in the observation process (although ‘specimen preservation could be considered a specific activity per se’).
* The physical characteristics of the features themselves are of little interest, except perhaps to the manager of a sampling campaign.
* Typically, the Sample is a Feature which is intended to be representative of a FeatureOfInterest on which Observations may be made. As such, it may carry a characteristic pertaining to the observedProperty being evaluated by the Observation.

EXAMPLE 1 A profile typically samples a water- or atmospheric-column; a well samples the water in an aquifer; a tissue specimen samples a part of an organism.

EXAMPLE 2 A statistical sample is often designed to be characteristic of an entire population, so that Observations can be made regarding the sample that provide a good estimate of the properties of the population.

### Association sampling

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sample/sampling-sem | The **Sampling** the **Sample** is the result of.  If **Sampling**(s)are described they SHALL be referred to using the association with the role **sampling**. |

### Association preparationStep

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sample/preparationStep-sem | The **PreparationStep**(s) applied to prepare the **Sample**.  If **PreparationSteps** are described they SHALL be referred to using the association with the role **preparationStep**. |

### Association sampledFeature

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sample/sampledFeature-sem | The **sampledFeature** is the featurethe **Sample** is intended to be representative of.  References to the sampled feature SHALL be provided using the association with the role **sampledFeature**. |

NOTE The sampled feature is usually a real-world feature from an application domain.

EXAMPLE 1 A profile typically samples a water or atmospheric column; a well samples the water in an aquifer; a tissue specimen samples a part of an organism.

EXAMPLE 2 A statistical sample is often designed to be characteristic of an entire population, so that Observations can be made regarding the sample that provide a good estimate of the properties of the population.

### Association relatedSample

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sample/relatedSample-sem | A **Sample** the **Sample** is related to.  If a reference to a related **Sample** is provided, the association with role **relatedSample** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

NOTE Sample are frequently related to each other, as parts of complexes, and in other ways.

EXAMPLE Sampling points are often located along a sampling curve; material samples are usually obtained from a sampling point; pixels are part of a scene; stations are often part of an array.

## Sampling

### Sampling Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/Sampling |
| Target type | Conceptual model |
| Name | Conceptual Sample - Sampling |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Requirement | /req/sam-cpt/Sampling/Sampling-sem |
| Requirement | /req/sam-cpt/Sampling/sample-sem |
| Requirement | /req/sam-cpt/Sampling/featureOfInterest-sem |
| Requirement | /req/sam-cpt/Sampling/featureOfInterest-card |
| Requirement | /req/sam-cpt/Sampling/sampler-sem |
| Requirement | /req/sam-cpt/Sampling/samplingProcedure-sem |
| Requirement | /req/sam-cpt/Sampling/relatedSampling-sem |
| Requirement | /req/obs-basic/gen/link-sem |

### Interface Sampling

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/Sampling-sem | **Sampling** is an act applying a **SamplingProcedure** to create or transform one or more **Sample**(s). |

EXAMPLES

* Crushing a rock sample in a ball mill.
* Digging a pit through a soil sequence.
* Dividing a field site into quadrants.
* Drawing blood from a patient.
* Extracting water from an observation well.
* Extracting a sample from a defined environmental monitoring station.
* Registering an image of the landscape.
* Sieving a powder to separate the subset finer than 100-mesh.
* Selecting a subset of a population.
* Splitting a piece of drill-core to create two new samples.
* Taking a diamond-drill core from a rock outcrop.

### Association sample

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/sample-sem | The **Sample** generated by the **Sampling.**  If **Samples** are described they SHALL be referred to using the association with the role **sample**. |

### Association featureOfInterest

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/featureOfInterest-sem | The concept, real-world object or phenomenon (feature-of-interest)the **Sample**(s)of the **Sampling** represent. |

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/featureOfInterest-card | Reference to the feature-of-interest SHALL be done using the association with the role **featureOfInterest**. |

### Association sampler

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/sampler-sem | The **Sampler** that performed the **Sampling**.  If **Sampler**(s)are described they SHALL be referred to using the association with the role **sampler**. |

### Association samplingProcedure

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/samplingProcedure-sem | The **SamplingProcedure** used by the **Sampling.**  If **SamplingProcedures** are described they SHALL be referred to using the association with the role **samplingProcedure**. |

### Association relatedSampling

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampling/relatedSampling-sem | Related **Sampling**(s).  If a reference to a related **Sampling** is provided, the association with role **relatedSampling** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

## Sampler

### Sampler Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/Sampler |
| Target type | Conceptual model |
| Name | Conceptual Sample - Sampler |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/sam-cpt/Sampler/Sampler-sem |
| Requirement | /req/sam-cpt/Sampler/sampling-sem |
| Requirement | /req/sam-cpt/Sampler/implementedProcedure-sem |
| Requirement | /req/obs-basic/gen/link-sem |

### Interface Sampler

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampler/Sampler-sem | A **Sampler** is a device or entity (including humans) that is used by, or implements, a **SamplingProcedure** to create or transform one or more **Sample**(s). |

EXAMPLES

* A ball mill, diamond drill, hammer.
* A hypodermic syringe and needle.
* An image sensor, a soil auger.
* A human being.

NOTE All the examples above can act as sampling devices (i.e., be Samplers). However, sometimes the distinction between the Sampler and the Sensor is not evident, as they are packaged as a unit. A Sampler need not be a physical device.

### Association sampling

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampler/sampling-sem | The **Sampling** act performed by the **Sampler**.  If **Sampling**(s)are described they SHALL be referred to using the association with the role **sampling**. |

### Association implementedProcedure

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/Sampler/implementedProcedure-sem | The **Procedure** implemented by the **Sampler**.  If **Procedure**(s)are described they SHALL be referred to using the association with the role **implementedProcedure**. |

## PreparationStep

### PreparationStep Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/PreparationStep |
| Target type | Conceptual model |
| Name | Conceptual Sample - PreparationStep |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Requirement | /req/sam-cpt/PreparationStep/PreparationStep-sem |
| Requirement | /req/sam-cpt/PreparationStep/processingDetails-sem |
| Requirement | /req/sam-cpt/PreparationStep/preparedSample-sem |
| Requirement | /req/obs-basic/gen/link-sem |

### Interface PreparationStep

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/PreparationStep/PreparationStep-sem | A **PreparationStep** is an individual step pertaining to a **PreparationProcedure**. |

### Association processingDetails

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/PreparationStep/processingDetails-sem | A **PreparationProcedure** step performed on the **Sample** the **PreparationStep** pertains to.  If **PreparationProcedure**(s)are described they SHALL be referred to using the association with the role **processingDetails**. |

### Association preparedSample

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/PreparationStep/preparedSample-sem | The **Sample** on which the **PreparationProcedure** is performed.  The **Sample** SHALL be referred to using the association with the role **preparedSample**. |

## PreparationProcedure

### PreparationProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/PreparationProcedure |
| Target type | Conceptual model |
| Name | Conceptual Sample - PreparationProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Procedure |
| Requirement | /req/sam-cpt/PreparationProcedure/PreparationProcedure-sem |
| Requirement | /req/sam-cpt/PreparationProcedure/samplePreparationStep-sem |
| Requirement | /req/obs-basic/gen/link-sem |

### Interface PreparationProcedure

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/PreparationProcedure/PreparationProcedure-sem | The description of preparation steps performed on a **Sample**. |

### Association samplePreparationStep

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/PreparationProcedure/samplePreparationStep-sem | If the **PreparingProcedure** provides information on the **PreparationStep** where this procedure has been used, the association with the role **samplePreparationStep** SHALL be used. |

## SamplingProcedure

### SamplingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-cpt/SamplingProcedure |
| Target type | Conceptual model |
| Name | Conceptual Sample - SamplingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/obs-cpt/Procedure |
| Requirement | /req/sam-cpt/SamplingProcedure/SamplingProcedure-sem |
| Requirement | /req/sam-cpt/SamplingProcedure/sampling-sem |
| Requirement | /req/sam-cpt/SamplingProcedure/sampler-sem |
| Requirement | /req/obs-basic/gen/link-sem |

### Interface SamplingProcedure

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/SamplingProcedure/SamplingProcedure-sem | The description of steps performed by a **Sampler** in order to extract a **Sample** from its **sampledFeature** in the frame of a **Sampling.** |

### Association sampling

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/SamplingProcedure/sampling-sem | If the **SamplingProcedure** provides information on the **Sampling** where this procedure has been used, the association with the role **sampling** SHALL be used. |

### Association sampler

|  |  |
| --- | --- |
| **Requirement** /req/sam-cpt/SamplingProcedure/sampler-sem | If the **SamplingProcedure** provides information on the **Sampler** that implements this procedure, the association with the role **sampler** SHALL be used. |

# Abstract Sample Core

## General

### Abstract Sample Core Package Requirements

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core |
| Target type | Logical model |
| Name | Abstract Sample core package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-core/AbstractSample |
| Imports | /req/sam-core/AbstractSampling |
| Imports | /req/sam-core/AbstractSampler |
| Imports | /req/sam-core/AbstractSamplingProcedure |
| Imports | /req/sam-core/AbstractPreparationProcedure |
| Imports | /req/sam-core/AbstractPreparationStep |

## AbstractSample

### AbstractSample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractSample |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractSample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-cpt/Sample |
| Imports | /req/obs-core/NamedValue |
| Requirement | /req/sam-core/AbstractSample/sampleType-sem |
| Requirement | /req/sam-core/AbstractSample/parameter-sem |
| Requirement | /req/obs-core/gen/metadata-sem |
| Requirement | /req/sam-core/AbstractSampleType/AbstractSampleType-sem |



**Figure 24 — Context diagram for Abstract Sample core — AbstractSample.**

### Attribute sampleType

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSample/sampleType-sem | The type of **Sample** according to a community agreed typology.  If information on the type of **AbstractSample** is provided, the attribute **sampleType:AbstractSampleType** shall be used. |

### Attribute parameter

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSample/parameter-sem | Arbitrary event-specific parameter relevant to the **Sample**.  If additional parameter information is provided, the property **parameter**:**NamedValue** SHALL be used. |

EXAMPLE When taking water samples, the sampling procedure specifies the amount of time that must pass to allow sediments to settle. As reality is rarely as exact as plans, the actual waiting time applied to a specific sample can be stored in the parameter.

NOTE Parameter should NOT be utilized to provide information already contained in the model by existing attributes or associations.

## AbstractSampling

### AbstractSampling Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractSampling |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractSampling |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Imports | /req/sam-cpt/Sampling |
| Imports | /req/obs-core/NamedValue |
| Requirement | /req/sam-core/AbstractSampling/samplingLocation-sem |
| Requirement | /req/sam-core/AbstractSampling/time-sem |
| Requirement | /req/sam-core/AbstractSampling/parameter-sem |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 25 — Context diagram for Abstract Sample core — AbstractSampling.**

### Attribute samplingLocation

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSampling/samplingLocation-sem | If location information pertaining to the **Sampling** is provided, the attribute **samplingLocation:Geometry** SHALL be used. |

### Attribute time

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSampling/time-sem | If information on the time of the **Sampling** is provided, the attribute **time:TM\_Object** SHALL be used. |

### Attribute parameter

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSampling/parameter-sem | Arbitrary event-specific parameter relevant to the **Sampling**.  If additional parameter information is provided, the property **parameter**:**NamedValue** SHALL be used. |

EXAMPLE When taking water samples, the sampling procedure specifies that an amount of time must pass to allow sediments to settle. The exact waiting time used in this Sampling can be stored in the parameter.

NOTE Parameter should NOT be utilized to provide information already contained in the model by existing attributes or associations.

## AbstractSampler

### AbstractSampler Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractSampler |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractSampler |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-cpt/Sampler |
| Requirement | /req/sam-core/AbstractSampler/samplerType-sem |
| Requirement | /req/obs-core/gen/metadata-sem |
| Requirement | /req/sam-core/AbstractSamplerType/AbstractSamplerType-sem |



**Figure 26 — Context diagram for the Abstract Sample core — AbstractSampler.**

### Attribute samplerType

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSampler/samplerType-sem | The type of **Sampler** according to a community agreed typology.  If information on the type of **AbstractSampler** is provided, the attribute **samplerType:AbstractSamplerType** shall be used. |

EXAMPLES

* A ball mill, diamond drill, hammer,
* A hypodermic syringe and needle,
* An image sensor, a soil auger,
* A human being.

## AbstractSamplingProcedure

### AbstractSamplingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractSamplingProcedure |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractSamplingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-cpt/SamplingProcedure |
| Requirement | /req/obs-core/gen/metadata-sem |



Figure 27 – Context diagram for Abstract Sample core – AbstractSamplingProcedure

## AbstractPreparationProcedure

### AbstractPreparationProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractPreparationProcedure |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractPreparationProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-cpt/PreparationProcedure |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 28 — Context diagram for Abstract Sample core —AbstractPreparationProcedure and AbstractPreparationStep.**

## AbstractPreparationStep

### AbstractPreparationStep Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-core/AbstractPreparationStep |
| Target type | Logical model |
| Name | Abstract Sample core - AbstractPreparationStep |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19108:2002 Geographic information – Temporal schema, Application schemas for data transfer conformance class |
| Imports | /req/sam-cpt/PreparationStep |
| Requirement | /req/sam-core/AbstractPreparationStep/description-sem |
| Requirement | /req/sam-core/AbstractPreparationStep/time-sem |
| Requirement | /req/obs-core/gen/metadata-sem |

### Attribute description

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractPreparationStep/description-sem | Description of the **preparationStep**.  If a description pertaining to the **preparationStep** is provided, the attribute **description:CharacterString** SHALL be used. |

### Attribute time

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractPreparationStep/time-sem | Time of the **preparationStep**.  If information on the time of the **preparationStep** of the **Sampling** is provided, the attribute **time:TM\_Object** SHALL be used. |

## Codelists

### AbstractSampleType

The code list AbstractSampleType can be specialized as required to firm up semantics of sample types.

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSampleType/AbstractSampleType-sem | An empty extension point for providing various classification schemes for **Samples**.  If **Sample** classification schemes are used in the implementing application schemas, a concrete realization SHALL be created for the application. |

### AbstractSamplerType

The code list AbstractSamplerType can be specialized as required to firm up semantics of sampler types.

|  |  |
| --- | --- |
| **Requirement** /req/sam-core/AbstractSamplerType/AbstractSamplerType-sem | An empty extension point for providing various classification schemes for **Samplers**.  If **Sampler** classification schemes are used in the implementing application schemas, a concrete realization SHALL be created for the application. |

# Basic Samples

## General

### Basic Samples Package Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic |
| Target type | Logical model |
| Name | Basic Samples package |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-basic/Sample |
| Imports | /req/sam-basic/SpatialSample |
| Imports | /req/sam-basic/MaterialSample |
| Imports | /req/sam-basic/StatisticalSample |
| Imports | /req/sam-basic/Sampling |
| Imports | /req/sam-basic/Sampler |
| Imports | /req/sam-basic/SamplingProcedure |
| Imports | /req/sam-basic/PreparationProcedure |
| Imports | /req/sam-basic/PreparationStep |
| Imports | /req/sam-basic/SampleCollection |
| Requirement | /req/sam-basic/SampleTypeByGeometryType/SampleTypeByGeometryType-sem |
| Requirement | /req/sam-basic/SampleTypeByGeometryType/SampleTypeByGeometryType-con |

## Sample

### Sample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/Sample |
| Target type | Logical model |
| Name | Basic Samples - Sample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-core/AbstractSample |



**Figure 29 — Context diagram for Basic Samples — Sample, SpatialSample, StatisticalSample and MaterialSample.**

## SpatialSample

### SpatialSample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/SpatialSample |
| Target type | Logical model |
| Name | Basic Samples - SpatialSample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19107:2019 Geographic information – Spatial schema, Geometry conformance class |
| Imports | /req/sam-basic/Sample |
| Requirement | /req/sam-basic/SpatialSample/SpatialSample-sem |
| Requirement | /req/sam-basic/SpatialSample/shape-sem |
| Requirement | /req/sam-basic/SpatialSample/horizontalPositionalAccuracy-sem |
| Requirement | /req/sam-basic/SpatialSample/verticalPositionalAccuracy-sem |

### Feature type SpatialSample

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SpatialSample/SpatialSample-sem | A **SpatialSample** is a geospatial **Sample**. |

NOTE When observations are made to estimate properties of a geospatial feature, in particular where the value of a property varies within the scope of the feature, a SpatialSample is used. Depending on accessibility and on the nature of the expected property variation, the SpatialSample may be extensive in one, two or three spatial dimensions.

EXAMPLE

* Typically an Observation ‘site’ or 'station' connotes the 'world in the vicinity of the site (or station)', so the observed properties relate to the physical medium at the station, and not to any physical artifact such as a mooring, buoy, benchmark, monument, well, etc.
* Some common names for SpatialSample used in various application domains include Borehole, Flightline, Interval, Lidar Cloud, Map Horizon, Microscope Slide, Mine Level, Mine, Observation Well, Profile, Pulp, Quadrat, Scene, Section, ShipsTrack, Spot, Station, Swath, Trajectory, Traverse, etc.

### Attribute shape

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SpatialSample/shape-sem | The **shape** is the Geometry of the **SpatialSample**.  If location information pertaining to the **SpatialSample** is provided, the attribute **shape:Geometry** SHALL be used. |

NOTE The shape of the SpatialSample is the context for domain decomposition.

EXAMPLE Logs of different properties along a well or borehole might use different intervals, and sub-samples might be either spatially instantaneous, or averaged in some way over an interval. The position of the samples can be conveniently described in terms of offsets in a linear coordinate reference system that is defined by the shape of the well axis.

### Attribute horizontalPositionalAccuracy

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SpatialSample/horizontalPositionalAccuracy-sem | The positional accuracyof the horizontal component of the Geometry of the **SpatialSample**.  If horizontal positional accuracyinformation pertaining to the **SpatialSample** is provided, the attribute **horizontalPositionalAccuracy:Any** SHALL be used. |

### Attribute verticalPositionalAccuracy

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SpatialSample/verticalPositionalAccuracy-sem | The positional accuracyof the vertical component of the Geometry of the **SpatialSample**.  If horizontal positional accuracyinformation pertaining to the **SpatialSample** is provided, the attribute **verticalPositionalAccuracy:Any** SHALL be used. |

## MaterialSample

### MaterialSample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/MaterialSample |
| Target type | Logical model |
| Name | Basic Samples - MaterialSample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Imports | /req/sam-basic/Sample |
| Imports | /req/sam-basic/PhysicalDimension |
| Imports | /req/sam-basic/NamedLocation |
| Requirement | /req/sam-basic/MaterialSample/MaterialSample-sem |
| Requirement | /req/sam-basic/MaterialSample/size-sem |
| Requirement | /req/sam-basic/MaterialSample/storageLocation-sem |
| Requirement | /req/sam-basic/MaterialSample/sourceLocation-sem |

### Feature type MaterialSample

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/MaterialSample/MaterialSample-sem | A **MaterialSample** is a physical, tangible **Sample**. |

NOTE 1 MaterialSamples that are curated and preserved are sometimes known as 'specimens'.

NOTE 2 MaterialSamples can be destroyed in connexion with the observation act.

NOTE 3 A MaterialSample is a physical Sample of a FeatureOfInterest, obtained for Observation(s) normally carried out ex-situ, sometimes in a laboratory.

### Attribute size

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/MaterialSample/size-sem | The **size** describes a physical extent of the **MaterialSample**.  If size information pertaining to the **MaterialSample** is provided, the attribute **size:PhysicalDimension** SHALL be used. |

NOTE The size may be length, mass, volume, etc., as appropriate for the MaterialSample instance and its material type.

### Attribute storageLocation

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/MaterialSample/storageLocation-sem | The **storageLocation** is the location of a **MaterialSample**.  If information pertaining to the storage location of the **MaterialSample** is provided, the attribute **storageLocation:NamedLocation** SHALL be used. |

NOTE The storageLocation may be a location such as a shelf in a warehouse or a drawer in a museum.

### Attribute sourceLocation

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/MaterialSample/sourceLocation-sem | The **sourceLocation** is the location from where the **MaterialSample** was obtained.  If information pertaining to the source location of the **MaterialSample** is provided, the attribute **sourceLocation:Geometry** SHALL be used. |

NOTES

- Where a MaterialSample has a relatedSample whose location provides an unambiguous location then this attribute is not required. However, if the specific sampling location within the sampledFeature is important, then the sourceLocation can be used to provide such location information.

- The attribute sourceLocation of the MaterialSample may not be necessary in the case the related Sampling act samplingLocation attribute is providedStatisticalSample.

## StatisticalSample

### StatisticalSample Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/StatisticalSample |
| Target type | Logical model |
| Name | Basic Samples - StatisticalSample |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-basic/Sample |
| Imports | /req/sam-basic/StatisticalClassification |
| Requirement | /req/sam-basic/StatisticalSample/StatisticalSample-sem |
| Requirement | /req/sam-basic/StatisticalSample/classification-sem |

### Feature type StatisticalSample

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/StatisticalSample/StatisticalSample-sem | A **StatisticalSample** is a statistical subset of a feature-of-interest, defined for the purpose of creating **Observation**(s). |

NOTE 1 StatisticalSamples usually apply to populations or other sets, of which certain subset may be of specific interest.

EXAMPLE The male or female subset of a population.

### Attribute classification

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/StatisticalSample/classification-sem | The **classification** describes a criterion by which the subset was defined.  If information pertaining to the subsetting criteria by which a **StatisticalSample** has been defined is provided, the attribute **classification:StatisticalClassification** SHALL be used. |

NOTE The classification may be age, gender, etc., as appropriate for the set or population on which the subsetting is performed.

## Sampling

### Sampling Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/Sampling |
| Target type | Logical model |
| Name | Basic Samples - Sampling |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-core/AbstractSampling |



**Figure 30 — Context diagram for Basic Samples — Sampling.**

## Sampler

### Sampler Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/Sampler |
| Target type | Logical model |
| Name | Basic Samples - Sampler |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Imports | /req/sam-core/AbstractSampler |



**Figure 31 — Context diagram for Basic Samples — Sampler.**

## SamplingProcedure

### SamplingProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/SamplingProcedure |
| Target type | Logical model |
| Name | Basic Samples - SamplingProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Imports | /req/sam-core/AbstractSamplingProcedure |
| Requirement | /req/obs-basic/gen/link-sem |



Figure 32 — (Informative) Included direct and indirect requirements and recommendations of the Basic Samples — SamplingProcedure requirements class.

## PreparationProcedure

### PreparationProcedure Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/PreparationProcedure |
| Target type | Logical model |
| Name | Basic Samples - PreparationProcedure |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Imports | /req/sam-core/AbstractPreparationProcedure |
| Requirement | /req/obs-basic/gen/link-sem |



Figure 33 — (Informative) Included direct and indirect requirements and recommendations of the Basic Samples — PreparationProcedure requirements class.

## PreparationStep

### PreparationStep Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/PreparationStep |
| Target type | Logical model |
| Name | Basic Samples - PreparationStep |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Imports | /req/sam-core/AbstractPreparationStep |
| Requirement | /req/obs-basic/gen/link-sem |



Figure 34 — (Informative) Included direct and indirect requirements and recommendations of the Basic Samples — PreparationStep requirements class.

## SampleCollection

### SampleCollection Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/SampleCollection |
| Target type | Logical model |
| Name | Basic Samples - SampleCollection |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Requirement | /req/sam-basic/SampleCollection/SampleCollection-sem |
| Requirement | /req/sam-basic/SampleCollection/member-sem |
| Requirement | /req/sam-basic/SampleCollection/relatedCollection-sem |
| Requirement | /req/obs-core/gen/metadata-sem |



**Figure 35 — Context diagram for Basic Samples — SampleCollection.**

### Feature type SampleCollection

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SampleCollection/SampleCollection-sem | A collection of **Sample**s. |

### Association member

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SampleCollection/member-sem | A **Sample** that is partof this **SampleCollection**.  If the **SampleCollection** has members, the association with the role **member** SHALL be used. |

### Association relatedCollection

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SampleCollection/relatedCollection-sem | A **SampleCollection** the **SampleCollection** is related to.  If a reference to a related **SampleCollection** is provided, the association with role **relatedCollection** SHALL be used. The **context:GenericName** qualifier of this association may be used to provide further information as to the nature of the relation. |

## PhysicalDimension

### PhysicalDimension Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/PhysicalDimension |
| Target type | Logical model |
| Name | Basic Samples - PhysicalDimension |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Requirement | /req/sam-basic/PhysicalDimension/PhysicalDimension-sem |
| Requirement | /req/sam-basic/PhysicalDimension/dimension-sem |
| Requirement | /req/sam-basic/PhysicalDimension/value-sem |

### Data type PhysicalDimension

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/PhysicalDimension/PhysicalDimension-sem | A dataType for the provision of various size quantities. |

### Attribute dimension

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/PhysicalDimension/dimension-sem | The name of the **PhysicalDimension** about which a **value** is provided.  The identifier of the physical dimension SHALL be provided in the attribute **dimension:URI.** |

### Attribute value

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/PhysicalDimension/value-sem | The **value** of the **PhysicalDimension**.  The measure of the quantity being provided SHALL be provided in the attribute **value:Measure** |

## NamedLocation

### NamedLocation Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/NamedLocation |
| Target type | Logical model |
| Name | Basic Samples - NamedLocation |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreTypes conformance class |
| Dependency | ISO 19107:2019 Geographic information — Spatial schema, Geometry conformance class |
| Requirement | /req/sam-basic/NamedLocation/NamedLocation-sem |
| Requirement | /req/sam-basic/NamedLocation/address-sem |
| Requirement | /req/sam-basic/NamedLocation/name-sem |
| Requirement | /req/sam-basic/NamedLocation/representativeGeometry-sem |

### Data type NamedLocation

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/NamedLocation/NamedLocation-sem | A location identified by its name, address, spatial geometry or a combination of any of these three. |

### Attribute address

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/NamedLocation/address-sem | An **address** used for identifying a **NamedLocation.**  If **address** information is provided, the attribute **address:Any** SHALL be used. |

### Attribute name

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/NamedLocation/name-sem | A **name** used for identifying a **NamedLocation.**  If **name** information is provided, the attribute **name:GenericName** SHALL be used. |

### Attribute representativeGeometry

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/NamedLocation/representativeGeometry-sem | A geometry used for providing a representative spatial location of a **NamedLocation.**  If **geometry** is provided, the attribute **representativeGeometry:Geometry** SHALL be used. |

## StatisticalClassification

### StatisticalClassification Requirements Class

|  |  |
| --- | --- |
| **Requirements Class** | /req/sam-basic/StatisticalClassification |
| Target type | Logical model |
| Name | Basic Samples - StatisticalClassification |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, UML2 conformance class |
| Dependency | ISO 19103:2015 Geographic information – Conceptual schema language, CoreExtendedTypes conformance class |
| Requirement | /req/sam-basic/StatisticalClassification/StatisticalClassification-sem |
| Requirement | /req/sam-basic/StatisticalClassification/concept-sem |
| Requirement | /req/sam-basic/StatisticalClassification/classification-sem |

### Data type StatisticalClassification

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/StatisticalClassification/StatisticalClassification-sem | A dataType for the provision of information on statistical classifications. |

### Attribute concept

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/StatisticalClassification/concept-sem | The **concept** by which a **StatisticalClassification** is to be performed.  The name of the concept by which the statistical classification is performed SHALL be provided in the attribute **concept:URI.** |

EXAMPLE The concept for a statistical classification could be age, gender, color, size etc.

### Attribute classification

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/StatisticalClassification/classification-sem | The explicit **classification class** pertaining to the classification concept described by the **StatisticalClassification**.  The classification class of the **StatisticalClassification** SHALL be provided in the attribute **classification:URI.** |

EXAMPLE The classification for a statistical classification could be:

* Age Brackets: [0-10], [10-20]
* Genders: Male, Female, Other
* Color: Red, Green, Blue

## Codelists

### SampleTypeByGeometryType

The code list SampleTypeByGeometryType is a specialization of AbstractSampleType created to support the legacy sample types from the previous version of this standard.

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SampleTypeByGeometryType/SampleTypeByGeometryType-sem | The following entries SHALL be provided:   * point: the provided geometry is of type Point. * curve: the provided geometry is of type Curve. * surface: the provided geometry is of type Surface. * solid: the provided geometry is of type Solid. |

|  |  |
| --- | --- |
| **Requirement** /req/sam-basic/SampleTypeByGeometryType/SampleTypeByGeometryType-con | The following constraints SHALL be applied to the value of the result association of the Observation based on the codelist value used:   * If value "point" is used, the provided geometry shall be of type Point. * If value "curve" is used, the provided geometry shall be of type Curve. * If value "surface" is used, the provided geometry shall be of type Surface. * If value "solid" is used, the provided geometry shall be of type Solid. |

1. (normative)  
     
   Abstract Test Suite
   1. Abstract tests for Conceptual Observation schema package
      1. Conceptual Observation schema package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt** |
| Requirements | /req/obs-cpt |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - Deployment

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/Deployment** |
| Requirements | /req/obs-cpt/Deployment |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - Host

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/Host** |
| Requirements | /req/obs-cpt/Host |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - ObservableProperty

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/ObservableProperty** |
| Requirements | /req/obs-cpt/ObservableProperty |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - Observation

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/Observation** |
| Requirements | /req/obs-cpt/Observation |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - Observer

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/Observer** |
| Requirements | /req/obs-cpt/Observer |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - ObservingProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/ObservingProcedure** |
| Requirements | /req/obs-cpt/ObservingProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Observation - Procedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-cpt/Procedure** |
| Requirements | /req/obs-cpt/Procedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* 1. Abstract tests for Abstract Observation core package
     1. Abstract Observation core package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core** |
| Requirements | /req/obs-core |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractDeployment

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractDeployment** |
| Requirements | /req/obs-core/AbstractDeployment |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractHost

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractHost** |
| Requirements | /req/obs-core/AbstractHost |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractObservableProperty

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractObservableProperty** |
| Requirements | /req/obs-core/AbstractObservableProperty |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractObservation

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractObservation** |
| Requirements | /req/obs-core/AbstractObservation |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractObservationCharacteristics

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractObservationCharacteristics** |
| Requirements | /req/obs-core/AbstractObservationCharacteristics |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractObserver

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractObserver** |
| Requirements | /req/obs-core/AbstractObserver |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - AbstractObservingProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/AbstractObservingProcedure** |
| Requirements | /req/obs-core/AbstractObservingProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Observation core - NamedValue

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-core/NamedValue** |
| Requirements | /req/obs-core/NamedValue |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* 1. Abstract tests for Basic Observations package
     1. Basic Observations package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic** |
| Requirements | /req/obs-basic |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - Deployment

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/Deployment** |
| Requirements | /req/obs-basic/Deployment |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - GenericDomainFeature

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/GenericDomainFeature** |
| Requirements | /req/obs-basic/GenericDomainFeature |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - Host

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/Host** |
| Requirements | /req/obs-basic/Host |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - ObservableProperty

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/ObservableProperty** |
| Requirements | /req/obs-basic/ObservableProperty |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - Observation

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/Observation** |
| Requirements | /req/obs-basic/Observation |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - ObservationCharacteristics

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/ObservationCharacteristics** |
| Requirements | /req/obs-basic/ObservationCharacteristics |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - ObservationCollection

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/ObservationCollection** |
| Requirements | /req/obs-basic/ObservationCollection |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - Observer

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/Observer** |
| Requirements | /req/obs-basic/Observer |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - ObservingCapability

|  |  |
| --- | --- |
| Conformance Class | /conf/obs-basic/ObservingCapability |
| Requirements | /req/obs-basic/ObservingCapability |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Observations - ObservingProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/obs-basic/ObservingProcedure** |
| Requirements | /req/obs-basic/ObservingProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* 1. Abstract tests for Conceptual Sample schema package
     1. Conceptual Sample schema package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt** |
| Requirements | /req/sam-cpt |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - PreparationProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/PreparationProcedure** |
| Requirements | /req/sam-cpt/PreparationProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - PreparationStep

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/PreparationStep** |
| Requirements | /req/sam-cpt/PreparationStep |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - Sample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/Sample** |
| Requirements | /req/sam-cpt/Sample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - Sampler

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/Sampler** |
| Requirements | /req/sam-cpt/Sampler |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - Sampling

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/Sampling** |
| Requirements | /req/sam-cpt/Sampling |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Conceptual Sample - SamplingProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-cpt/SamplingProcedure** |
| Requirements | /req/sam-cpt/SamplingProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* 1. Abstract tests for Abstract Sample core package
     1. Abstract Sample core package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core** |
| Requirements | /req/sam-core |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractPreparationProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractPreparationProcedure** |
| Requirements | /req/sam-core/AbstractPreparationProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractPreparationStep

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractPreparationStep** |
| Requirements | /req/sam-core/AbstractPreparationStep |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractSample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractSample** |
| Requirements | /req/sam-core/AbstractSample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractSampler

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractSampler** |
| Requirements | /req/sam-core/AbstractSampler |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractSampling

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractSampling** |
| Requirements | /req/sam-core/AbstractSampling |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Abstract Sample core - AbstractSamplingProcedure

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-core/AbstractSamplingProcedure** |
| Requirements | /req/sam-core/AbstractSamplingProcedure |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* 1. Abstract tests for Basic Samples package
     1. Basic Samples package

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic** |
| Requirements | /req/sam-basic |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - MaterialSample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/MaterialSample** |
| Requirements | /req/sam-basic/MaterialSample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - NamedLocation

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/NamedLocation** |
| Requirements | /req/sam-basic/NamedLocation |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - PhysicalDimension

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/PhysicalDimension** |
| Requirements | /req/sam-basic/PhysicalDimension |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - Sample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/Sample** |
| Requirements | /req/sam-basic/Sample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - SampleCollection

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/SampleCollection** |
| Requirements | /req/sam-basic/SampleCollection |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - Sampler

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/Sampler** |
| Requirements | /req/sam-basic/Sampler |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - Sampling

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/Sampling** |
| Requirements | /req/sam-basic/Sampling |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - SpatialSample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/SpatialSample** |
| Requirements | /req/sam-basic/SpatialSample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - StatisticalClassification

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/StatisticalClassification** |
| Requirements | /req/sam-basic/StatisticalClassification |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

* + 1. Basic Samples - StatisticalSample

|  |  |
| --- | --- |
| **Conformance Class** | **/conf/sam-basic/StatisticalSample** |
| Requirements | /req/sam-basic/StatisticalSample |
| Test purpose | Verify that all requirements from the requirements class have been fulfilled. |
| Test method | Inspect the documentation of the application, schema or profile. |
| Test Type | Capability |

1. (informative)  
     
   Common usage of OMS concepts
   1. Introduction

This International Standard defines concepts in support of a generic, cross-domain model for observations, measurements and samples. Concepts are taken from a variety of disciplines. The concepts are used within the model described in this standard in a consistent manner, but in order to achieve internal consistency, this varies from how the same concepts are used in some application domains. In order to assist in the correct application of the model across domains, this annex provides a mapping from Observations, measurements and samples (OMS) concepts to those used within some domains.

* 1. Earth Observations (EO)

**Table B.1 — Earth Observations (EO)**

|  |  |  |
| --- | --- | --- |
| **OMS** | **EO** | **Example** |
| Observation::result | Observation value, measurement value, observation | 35 µg/m3 |
| Observation::procedure | Method, sensor | ASTER, U.S. EPA Federal Reference Method for PM 2.5 |
| Observation::observedProperty | Parameter, variable | Reflectance, Particulate Matter 2.5 |
| Observation::proximateFeatureOfInterest:SpatialSample | 2-D swath or scene | Sampling grid |
| SpatialSample:sampledFeature | Earth surface | http://sweetontology.net/realm/PlanetarySurface |
| Observation::ultimateFeatureOfInterest | Earth surface | http://sweetontology.net/realm/PlanetarySurface |
| Observation::proximateFeatureOfInterest:SpatialSample | 3-D sampling space | Sampling grid |
| SpatialSample::sampledFeature | Media (air, water, …), Global Change Master Directory "Topic" | Troposphere |
| Observation::ultimateFeatureOfInterest | Media (air, water, …), Global Change Master Directory "Topic" | Troposphere |

* 1. Metrology

**Table B.2 — Metrology**

|  |  |  |
| --- | --- | --- |
| **OMS** | **Metrology** | **Example: mass measurement** |
| Observation::result | Value | 35 mg |
| Observation::procedure | Instrument | Balance |
| Observation::observedProperty | Measurand | Mass |

* 1. Earth science simulations

**Table B.3 — Earth science simulations**

|  |  |
| --- | --- |
| **OMS** | **Earth science** |
| Observation::result | A model or field |
| Observation::observedProperty | Variable, parameter |
| Observation::proximateFeatureofInterest:SpatialSample | Section, swath, volume, grid |
| Observation::proximateFeatureofInterest:SpatialSample::sampledFeature | Atmosphere, ocean, solid earth |
| Observation::ultimateFeatureofInterest | Atmosphere, ocean, solid earth |
| Observation::procedure | Earth process simulator |
| Observation::phenomenonTime | Future date (forecasts), past date (hindcasts) |
| Observation::resultTime | Simulator execution date |
| Observation::validTime | Period when result is intended to be used |

* 1. Assay/Chemistry

**Table B.4 — Assay/Chemistry**

|  |  |
| --- | --- |
| **OMS** | **Geochemistry** |
| Observation::proximateFeatureOfInterest:MaterialSample | Sample |
| MaterialSample::sampledFeature:GeologicUnit | Ore body, Geologic Unit |
| MaterialSample::relatedSample:MaterialSample | Pulp, separation |
| MaterialSample::preparationStep | Sample preparation process |
| MaterialSample::sampling:Sampling:samplingProcedure | Sample collection process |
| MaterialSample::sourceLocation | Sample collection location |
| MaterialSample::size | Mass, length |
| MaterialSample::storageLocation | Store location |
| MaterialSample::sampling:Sampling:time | Sample collection date |
| Observation::phenomenonTime | Sample collection date |
| Observation::resultTime | Analysis date |
| Observation::result | Analysis |
| Observation::observedProperty | Analyte |
| Observation::procedure | Instrument, analytical process |

* 1. Geology field observations

**Table B.5 —** **Geology field observations**

|  |  |
| --- | --- |
| **OMS** | **Geology** |
| Observation::proximateFeatureOfInterest:SampleCollection | Outcrop |
| SampleCollection::member:SpatialSample | Location of structure observation |
| SpatialSample::sampledFeature:GeologicUnit | Geologic Unit |
| Observation::phenomenonTime | Outcrop visit date |
| Observation::observedProperty | Strike and dip, lithology, alteration state, etc. |
| SampleCollection::member:MaterialSample | Rock sample |
| MaterialSample::sampledFeature:GeologicUnit | Ore body, Geologic Unit |

* 1. Geotechnics observations

**Table B.6 — Geotechnics observations**

|  |  |
| --- | --- |
| **OMS** | **Geotechnical in situ test** |
| Observation::result | A log |
| Observation::observedProperty | A soil property (eg. gamma ray, resistivity, sound speed propagation) |
| Observation::proximateFeatureofInterest:SpatialSample | The borehole trajectory |
| Observation::proximateFeatureofInterest:SpatialSample::sampledFeature | A part of the Earth |
| Observation::ultimateFeatureofInterest | A part of the Earth |
| Observation::procedure | Geotechnical test procedure |
| Observation::phenomenonTime | Date and time of the test |
| Observation::resultTime | Date and time of the test |
| Observation::validTime | Date and time of the test |

* 1. Water quality observations

**Table B.7 — Water quality observations**

|  |  |
| --- | --- |
| **OMS** | **Water quality** |
| Observation::proximateFeatureOfInterest:SpatialSample | Water quality station at Cénac (France) |
| SpatialSample::sampledFeature:WaterBody | River (e.g. the Dordogne river) |
| SpatialSample::relatedSample:MaterialSample | Water Sample as sampled on-site |
| MaterialSample::sampledFeature:WaterBody | River (e.g. the Dordogne river) |
| MaterialSample::relatedSample:MaterialSample | Filtered sample (sub-sample of the initial one) |
| MaterialSample::sampledFeature:MaterialSample | The initial water sample that was sub-sampled |
| MaterialSample::preparationStep | Sample preparation process |
| MaterialSample::sampling:Sampling:samplingProcedure | Sample collection process |
| MaterialSample::sourceLocation | Sample collection location |
| MaterialSample::size | Volume of the water sampled |
| MaterialSample::storageLocation | Store location |
| MaterialSample::sampling:Sampling:time | Sample collection date |
| Observation::phenomenonTime | Sample collection date |
| Observation::resultTime | Analysis date |
| Observation::result | Analysis |
| Observation::observedProperty | Analyte (Nitrates, Phosphates …) |
| Observation::procedure | Instrument, analytical process (e.g. NF EN ISO 13395 Octobre 1996 / T90-012) |

* 1. Soil quality observations

**Table B.8 — Soil quality observations**

|  |  |
| --- | --- |
| **OMS** | **Soil quality** |
| Observation::proximateFeatureOfInterest:MaterialSample | A sub sample or the initial soil sample |
| MaterialSample::relatedSample:MaterialSample | Soil sample (can be a drilling core) |
| MaterialSample:relatedSample:SpatialSample | The borehole that was drilled and the core extracted from |
| Observation::ultimateFeatureOfInterest | Part of the lithosphere |
| MaterialSample::preparationStep | Sample preparation process |
| MaterialSample::sampling:Sampling:samplingProcedure | How the sample was collected or prepared |
| MaterialSample::sourceLocation | Where the sample has been collected |
| MaterialSample::storageLocation | Where the sample is stored |
| MaterialSample::sampling:Sampling:time | When the sample was collected |
| Observation::phenomenonTime | Sample collection date |
| Observation::resultTime | Analysis date |
| Observation::result | The result of the analysis |
| Observation::observedProperty | The analysed property (generally concentration of a constituent) |
| Observation::procedure | The analysis method |

1. (informative)  
     
   Changes in the Observation and Sample models  
   between ISO 19156:2011, edition 1 and ISO 19156:2021, edition 2

This annex contains information about the changes made in the Observation, Sampling and Specimen models between Observations and Measurements v2.0 (ISO 19156:2011, edition 1) and Observations, Measurements and Samples v3.0 (ISO 19156:2020, edition 2). It is intended for readers familiar with the O&M v2.0 and ISO 19156:2011 and provides detailed migration guidance for information systems and application schemas based on the O&M concepts.

* 1. Package and requirements class structure

The following UML packages were defined in the ISO 19156 Edition 1 (2011):

* Observation schema
  + observation <<ApplicationSchema>>
  + measurement <<ApplicationSchema>>
  + categoryObservation <<RequirementsClass>>
  + countObservation <<RequirementsClass>>
  + truthObservation <<RequirementsClass>>
  + temporalObservation <<RequirementsClass>>
  + geometryObservation <<RequirementsClass>>
  + complexObservation <<RequirementsClass>>
  + coverageObservation <<RequirementsClass>>
  + pointCoverageObservation <<RequirementsClass>>
  + timeSeriesObservation <<RequirementsClass>>
* Sampling Features
  + samplingFeature <<ApplicationSchema>>
  + spatialSamplingFeature <<ApplicationSchema>>
  + samplingPoint <<RequirementsClass>>
  + samplingCurve <<RequirementsClass>>
  + samplingSurface <<RequirementsClass>>
  + samplingSolid <<RequirementsClass>>
  + specimen <<RequirementsClass>>
* Domain specific sampling features <<informative>>
* Examples <<informative>>
* Sampling Coverage Observation <<informative>>
* General Feature Instance <<RequirementsClass>>
* Temporal Coverage <<RequirementsClass>>

The following conformance classes and the included Abstract Test Suite clauses were defined for Application Schemas in ISO 19156 Edition 1:

* Generic observation interchange: A.1.1
* Measurement interchange: A.1.1, A.1.2
* Category observation interchange: A.1.1, A.1.3
* Count observation interchange: A.1.1, A.1.4
* Truth observation interchange: A.1.1, A.1.5
* Temporal observation interchange: A.1.1, A.1.6
* Geometry observation interchange: A.1.1, A.1.7
* Complex observation interchange: A.1.1, A.1.8
* Discrete coverage observation interchange: A.1.1, A.1.9
* Point coverage observation interchange: A.1.1, A.1.10
* Time series observation interchange: A.1.1, A.1.11
* Sampling feature interchange: A.2.1, A.2.2
* Spatial sampling feature interchange: A.2.1 to A.2.3
* Sampling point interchange: A.2.1 to A.2.4
* Sampling curve interchange: A.2.1 to A.2.3, A.2.5
* Sampling surface interchange: A.2.1 to A.2.3, A.2.6
* Sampling solid interchange: A.2.1 to A.2.3, A.2.7
* Specimen interchange: A.2.1 to A.2.3, A.2.8

In ISO 19156 Edition 2 (2020) the UML packages were restructured to describe three levels of abstraction (conceptual schema, abstract core application schema and basic application schema) for both Observations and Samples:

* Conceptual Observation schema
* Conceptual Sample schema
* Abstract Observation core <<ApplicationSchema>>
* Abstract Sample core <<ApplicationSchema>>
* Basic Observations <<ApplicationSchema>>
* Basic Samples <<ApplicationSchema>>
* Examples <<informative>>
* Codelist realizations <<informative>>

The requirements classes of the ISO 19156 Edition 2 are much more fine-grained than in the conformance classes in Edition 1: There are several requirements classes considering the class or interface classifiers defined in each package, each capturing an atomic part of the package, typically a single class with its attributes and associations. Additionally, there are package-specific compound requirements classes that consist of requirements and recommendations for all classifiers defined of each package. For each of the requirements class there is a corresponding conformance class that a system may declare conformance to. Thus, the number of conformance classes in ISO 19156 Edition 2 (53 conformance classes) is much bigger than in the Edition 1 (18 conformance classes). For the complete list of Edition 2 conformance classes see Annex A.

* 1. Interfaces in the conceptual schema packages

The conceptual schema packages define terminology and semantics of the fundamental concepts of the OMS model using interfaces only, and provide the basis for fine-grained, isolated conformance class structure enabling classes in application schemas to associate themselves with any implementing class of the targeted interfaces.

Both the packages Conceptual Observation schema and Conceptual Sample schema consist of only interfaces with the attributes and associations of the essential concepts defined in the OMS standard. All the interfaces in these two packages are new to the ISO 19156 Edition 2, although most of them capture concepts defined in Edition 1 as classes.

There are a few completely new concepts added in the Edition 2:

* Observer (generator of an Observation events)
* Deployment (assignment of an Observation to a Host)
* Host (grouping of Observations, such as a physical platform, observing station or an observing campaign)
* Sampler (device or entity creating or transforming Samples)
* ObservationCollection (a collection of similar Observations)

Some of the following concepts have been renamed and/or partly redefined:

* OM\_Observation concept is now captured as the Observation interface.
* OM\_Process concept is now captured as the Procedure interface and its specializations ObservingProcedure, SamplingProcedure and PreparationProcedure.
* SF\_SamplingFeature concept is now captured as the Sample interface.
* A generic feature type instance defined in Edition 1 as GFI\_Feature as the target of the featureOfInterest association of the OM\_Observation and sampledFeature association of the SF\_SamplingFeature is removed and replaced with Any interface from ISO 19103 in favour of broadening the scope of these associations from feature objects as defined in ISO 19109 into any observed or sampled object.
* The metaclass GF\_PropertyType defined for describing the observed properties in Edition 1 has been removed and is now captured by the ObservableProperty interface.
* Sampling event information partly captured by SF\_Specimen attributes samplingTime, samplingMethod and samplingLocation in Edition 1 is now captured as the Sampling interface.
* Association class PreparationStep for describing the processingDetails association role from SF\_Specimen to SF\_Process in Edition 1 has been remodelled as an interface PreparationStep with the processingDetails association role to the PreparationProcedure interface.
  1. Realizations of the conceptual schemas as abstract and concrete feature type classes

The Abstract Observation core and the Abstract Sample core packages bind the interface concepts of the conceptual schemas with the ISO 19109 feature concept, and introduces these concepts and some related classes using the FeatureType stereotype and with more detailed set of attributes. They also introduce a mechanism for indirect FeatureType associations via corresponding conceptual schema interfaces providing a degree of conformance statement isolation: Implementations may choose to directly use some but not all of the FeatureType classes in the core or the basic packages, and still implement some of their associations using existing or bespoke domain classes as long as they conceptually and pertaining to their data content realize the corresponding interfaces.

While the Abstract Observation and Abstract Sample core packages provide a common basis for all ISO 19109 based implementations of the OMS conceptual model, the Basic Observations and the Basic Samples packages are designed as ready-to-use concrete implementations for these concepts. It is expected that the Basic package classes are used as a toolbox for implementing observations and samples related application schemas in pick-and-mix style: If the Basic package classes provide all necessary information for data management and exchange use cases in a particular application domain, they can simply be adopted as-is. In cases where more expressiveness is required, the Abstract core or the Basic package classes may be specialized as required without breaking the integrity of the model.

* 1. Modelling of the Observation concept
     1. OM\_Observation in Edition 1

The Observation concept was modelled as OM\_Observation class in Edition 1 as follows:

"An observation is an act that results in the estimation of the value of a feature property, and involves application of a specified procedure, such as a sensor, instrument, algorithm or process chain. [...]"

It had the following attributes, associations and cardinalities:

* featureOfInterest (Domain): GFI\_Feature [1]
* observedProperty (Phenomenon): GF\_PropertyType [1]
* procedure (ProcessUsed): OM\_Process [1]
* phenomenonTime: TM\_Object [1]
* resultTime: TM\_Instant [1]
* result (Range): Any [1]
* resultQuality: DQ\_Element [0..\*]
* parameter: NamedValue [0..\*]
* validTime: TM\_Period [0..1]
* relatedObservation: OM\_Observation [0..\*]
* metadata (Metadata): MD\_Metadata [0..1]

OM\_Observation had the following constraints:

* a parameter.name shall not appear more than once
* observedProperty shall be a phenomenon associated with the feature-of-interest
* procedure shall be suitable for observedProperty
* result type shall be suitable for observedProperty
  + 1. Observation in Edition 2

In Edition 2 Observation concept is modelled using one interface and three classes:

* Observation interface in the Conceptual Observation schema package,
* AbstractObservationCharacteristics in the Abstract Observation core package.
* AbstractObservation class in the Abstract Observation core package, and
* Observation class in the Basic Observations package.

The Observation interface is defined as follows:

"an act carried out by an observer to determine the value of an observable property of an object (feature-of-interest) by using a procedure; the value is provided as the result."

It has the following attributes, associations and cardinalities:

* featureOfInterest (Domain): Any [1..\*]
* observingProcedure: ObservingProcedure [1]
* observedProperty: ObservableProperty [1]
* observer: Observer [0..\*]
* host: Host [0..\*]
* phenomenonTime: TM\_Object [1]
* resultTime: TM\_Object [1]
* result (Range): Any [1]
* validTime: TM\_Period [0..\*]
* relatedObservation: Observation [0..\*]

The Observation interface contains the following constraints:

* at least one of either observer or host should be provided
* observedProperty should be a phenomenon associated with the featureOfInterest
* procedure should be suitable for the associated observedProperty
* result type should be suitable for the associated observedProperty

The AbstractObservationCharacteristics class describes common characteristics of Observations. Thus, in addition to serving as the base class for realizations of the Observation interface, it can also be utilized for the description of sets of related or similar Observations, as well as describing the observing capabilities of facilities hosting various observation devices. To enable such additional functionality, the cardinalities of the properties of the AbstractObservationCharacteristics has been relaxed to 0..\*.

AbstractObservationCharacteristics class has the following attributes, associations and cardinalities:

* ultimateFeatureOfInterest (Domain): Any [0..\*]
* proximateFeatureOfInterest (DomainProxy): Any [0..\*]
* observingProcedure: Conceptual Observation schema: ObservingProcedure [0..\*]
* observedProperty: Conceptual Observation schema: ObservableProperty [0..\*]
* observer: Conceptual Observation schema: Observer [0..\*]
* host: Conceptual Observation schema: Host [0..\*]
* phenomenonTime: TM\_Object [0..\*]
* resultTime: TM\_Object [0..\*]
* result (Range): Any [0..\*]
* resultQuality: Any [0..\*]
* parameter: NamedValue [0..\*]
* validTime: TM\_Object [0..\*]
* observationType: AbstractObservationType [0..\*]
* metadata: Any [0..\*]

AbstractObservation class specializes the AbstractObservationCharacteristics by realizing the Observation interface of the Conceptual Observation schema including the relatedObservation association, and by adding the following constraints:

* at least one proximateFeatureOfInterest or ultimateFeatureOfInterest shall be given
* attribute and association values shall be aligned with the observationType
* exactly one observedProperty shall be given
* exactly one phenomenonTime shall be given
* exactly one observingProcedure shall be given
* exactly one result shall be given
* exactly one resultTime shall be given
* observedProperty should be a phenomenon associated with the ultimateFeatureOfInterest or the proximateFeatureOfInterest
* parameter.name shall not appear more than once
* resultTime shall be of type TM\_Instant

The Observation class in the Basic Observations package is a concrete class specializing the AbstractObservation without any additional attributes, associations or constraints.

Considering the constraints defined in the AbstractObservation class, the Observation class in Edition 2 has the following properties with effective cardinalities and types (changes from Edition 1 in bold):

* **ultimateFeatureOfInterest: Any [0..\*] (1..\* if the cardinality of the proximateFeatureOfInterest is 0)**
* **proximateFeatureOfInterest: Any [0..\*] (1..\* if the cardinality of the ultimateFeatureOfInterest is 0)**
* **observingProcedure: Conceptual Observation schema: ObservingProcedure [1]**
* observedProperty: **Conceptual Observation schema: ObservableProperty [1]**
* **observer: Conceptual Observation schema: Observer [0..\*]**
* **host: Conceptual Observation schema: Host [0..\*]**
* phenomenonTime: TM\_Object [1]
* resultTime: TM\_Instant [1]
* result: Any [1]
* resultQuality: **Any** [0..\*]
* parameter: NamedValue [0..\*]
* validTime: TM\_Period [0..\*]
* **observationType: AbstractObservationType [0..\*]**
* metadata: **Any [0..\*]**
  + 1. Migration from OM\_Observation to Observation

An instance of the OM\_Observation class of Edition 1 can be expressed as an instance of the Observation class of the Basic Observations package as follows:

* OM\_Observation.featureOfInterest: GFI\_Feature becomes either Observation.ultimateFeatureOfInterest: Any or Observation.proximateFeatureOfInterest: Any depending on whether it represents the entity that is ultimately of interest in the act of observing or its proxy. Refactoring of the domain models may be required to separate the ultimate and proximate features of interest.
* OM\_Observation.observedProperty: GF\_PropertyType becomes the Observation.observedProperty: ObservableProperty.
* OM\_Observation.procedure: OM\_Process becomes either the Observation.observingProcedure: ObservingProcedure or Observation.observer: Observer depending on whether it describes the kind of the observing procedure (method) or an identifiable entity that generates the Observations (such as an individual sensor device). Refactoring of the domain models may be required to separate the observing procedure from the observer.
* OM\_Observation.phenomenonTime: TM\_Object becomes Observation. phenomenonTime: TM\_Object.
* OM\_Observation.resultTime: TM\_Instant becomes Observation.resultTime: TM\_Instant.
* OM\_Observation.result: Any becomes Observation.result: Any
* OM\_Observation.resultQuality: DQ\_Element becomes Observation.resultQuality: Any
* OM\_Observation.parameter: NamedValue becomes Observation.parameter: NamedValue
* OM\_Observation.validTime: TM\_Period becomes Observation.validTime: TM\_Period
* OM\_Observation.relatedObservation: OM\_Observation becomes Observation.relatedObservation: Observation
* OM\_Observation.metadata: MD\_Metadata [0..1] becomes Observation.metadata: Any

For information about transitioning the specialized Observation types of Edition 1 see the "Hard-typing vs. soft typing and codelist use" section below.

The **Table 8** summarizes the Observation mappings from the edition 2 Basic Observations package to edition 1.

**Table 8 — Observation mapping from 19156 edition 2 to edition 1 (informative)**

|  |  |  |
| --- | --- | --- |
| **Edition 2 class / property, Basic Observations package** | **Relation** | **Edition 1 class / property** |
| Observation | equivalent class | OM\_Observation |
| Observation.parameter | equivalent property | OM\_Observation.parameter |
| Observation.phenomenonTime | equivalent property | OM\_Observation.phenomenonTime |
| Observation.resultQuality | equivalent property | OM\_Observation.resultQuality |
| Observation.resultTime | equivalent property | OM\_Observation.resultTime |
| Observation.validTime | equivalent property | OM\_Observation.validTime |
| Observation.result | equivalent property | OM\_Observation.result |
| Observation.ultimateFeatureOfInterest | sub-property of | OM\_Observation.featureOfInterest |
| Observation.proximateFeatureOfInterest | sub-property of | OM\_Observation.featureOfInterest |
| Observation.observedProperty | equivalent property | OM\_Observation.observedProperty |
| Observation.observer | sub-property of | OM\_Observation.procedure |
| Observation.procedure | sub-property of | OM\_Observation.procedure |
| Observation.metadata | equivalent property | OM\_Observation.metadata |
| Observation.relatedObservation | equivalent property | OM\_Observation.relatedObservation |
| ObservingProcedure | equivalent class | OM\_Process |
| ObservableProperty | equivalent class | GF\_PropertyType |
| Observer |  | (no match) |
| Deployment |  | (no match) |
| Host |  | (no match) |

* 1. Modelling of the Sample and Sampling concepts
     1. SF\_SamplingFeature, SF\_Specimen SF\_SpatialSamplingFeature and in Edition 1

The Samping Feature concept was modelled as SF\_SamplingFeature class in Edition 1 as follows:

"Sampling features are artefacts of an observational strategy, and have no significant function outside of their role in the observation process. The physical characteristics of the features themselves are of little interest, except perhaps to the manager of a sampling campaign."

It had the following attributes, associations and cardinalities:

* sampledFeature (Intention): GFI\_Feature [1..\*]
* relatedSamplingFeature: SF\_SamplingFeature [0..\*], with association class SamplingFeatureComplex
* relatedObservation: OM\_Observation [0..\*]
* lineage: LI\_Lineage [0..1]
* parameter: NamedValue [0..\*]

The SF\_SamplingFeature was specialized by two sub-classes SF\_Specimen and SF\_SpatialSamplingFeature, the latter of which specialized further by their geometry type as SF\_SamplingPoint, SF\_SamplingCurve, SF\_SamplingSurface and SF\_SamplingSolid classes.

The SF\_Specimen was defined as follows:

"A Specimen is a physical sample, obtained for observation(s) carried out ex situ, sometimes in a laboratory."

It added the following attributes, associations and cardinalities to the SF\_SamplingFeature:

* processingDetails: SF\_Process [0..\*] with association class PreparationStep
* currentLocation: Location [0..1]
* materialClass: GenericName [1]
* samplingLocation: GM\_Object [0..1]
* samplingMethod: SF\_Process [0..1]
* samplingTime: TM\_Object [1]
* size: Measure [0..1]
* specimenType: GenericName [0..1]

The SF\_SpatialSamplingFeature was defined as follows:

"When observations are made to estimate properties of a geospatial feature, in particular where the value of a property varies within the scope of the feature, a spatial sampling feature is used."

It added the following attributes, associations and cardinalities to the SF\_SamplingFeature:

* hostedProcedure (Platform): OM\_Process [0..\*]
* positionalAccuracy: DQ\_PositionalAccuracy [0..2]
* shape: GM\_Object [1]

The sub-classes SF\_SamplingPoint, SF\_SamplingCurve, SF\_SamplingSurface and SF\_SamplingSolid did not add any attributes or associations, but override the shape association to point to GM\_Point, GM\_Curve, GM\_Surface and GM\_Solid respectively.

* + 1. Sample, SpatialSample, MaterialSample and StatisticalSample in Edition 2

Edition 2 introduces the Sample concept which is modelled using one interface and five classes:

* Sample interface in the Conceptual Sample schema package,
* AbstractSample class in the Abstract Sample core package, and
* Sample class and its specializations in the Basic Samples package:
  + SpatialSample class
  + StatisticalSample class, and
  + MaterialSample class.

The Sample interface is defined as follows:

"an object that is representative of a concept, real-world object or phenomenon."

It has the following attributes, associations and cardinalities:

* sampledFeature: Any [1..\*]
* relatedObservation: Conceptual Observation schema: Observation [0..\*]
* preparationStep: PreparationStep [0..\*]
* sampling: Sampling [0..\*]
* relatedSample: Sample [0..\*]

The AbstractSample class realizes the Sample interface as a feature type. It has the following attributes, associations and cardinalities:

* sampledFeature: Any [1..\*]
* relatedObservation: Conceptual Observation schema: Observation [0..\*]
* preparationStep: Conceptual Sample schema: PreparationStep [0..\*]
* sampling: Conceptual Sample schema: Sampling [0..\*]
* relatedSample: Conceptual Sample schema: Sample [0..\*]
* sampleType: AbstractSampleType [0..\*]
* parameter: NamedValue [0..\*]
* metadata: Any [0..\*]

The Sample class in the Basic Samples package is a concrete class specializing the AbstractSample without any additional attributes, associations or constraints. Its sub-classes add specialized properties to describe their particular characteristics:

* SpatialSample adds the following attributes:
  + shape: Geometry [0..1]
  + horizontalPositionalAccuracy: Any [0..1]
  + verticalPositionalAccuracy: Any [0..1]
* StatisticalSample adds the following attribute:
  + classification: StatisticalClassification [0..\*]
* MaterialSample adds the following attributes:
  + size: PhysicalDimension [0..\*]
  + sourceLocation: Geometry [0..1]
  + storageLocation: NamedLocation [0..1]
    1. Modelling of environmental monitoring stations

Note that in Edition 1 the SF\_SampingPoint class is associated with the concept of an environmental monitoring facility by the use of term "station":

"A common mode of sampling is at a point. In environmental measurements and monitoring the term Station is often used."

A related note is provided for the SF\_SpatialSamplingFeature.hostedProcedure:

"A common role for a spatial sampling feature is to host instruments or procedures deployed repetitively or permanently. If present, the association Platform shall link the SF\_SpatialSamplingFeature to an OM\_Process deployed at it. The OM\_Process has the role hostedProcedure with respect to the sampling feature."

The Sample (or SpatialSample) concept of the Edition 2 is not used for describing environmental monitoring stations and other entities generating Observations or hosting instruments. Instead they are modelled using the new Observer concept, which may be related to the Host concept via the Deployment concept. An environmental measurement station would be modelled as an instance of the Host class in the Basic Observations package or another domain-specific realization of the Host interface. An instrument, sensor or device hosted by the Station would be modelled by the Observer class in the Basic Observations package or another domain-specific realization of the Observer interface. The characteristics of the hostings or attachments of an Observer to its Hosts are described using the associated Deployment concept. The description of the observing procedures available for the specific Observer would be provided through the Observer.observingProceducedure: ObservingProcedure association.

* + 1. Migration from SF\_SamplingFeature to Sample

An instance of SF\_SamplingFeature class of Edition 1 can be expressed as an instance of the Sample class of the Basic Samples package as follows:

* SF\_SamplingFeature.sampledFeature: GFI\_Feature becomes Sample.sampledFeature: Any.
* SF\_SamplingFeature.relatedSamplingFeature: SF\_SamplingFeature becomes Sample.relatedSample: Conceptual Sample schema: Sample; the value role:GenericName attribute of association class SamplingFeatureComples becomes the value of the context:GenericName qualifier of the relatedSample association.
* SF\_SamplingFeature.relatedObservation: OM\_Observation becomes Sample.relatedObservation: Conceptual Sample schema: Observation.
* SF\_SamplingFeature.lineage: LI\_Lineage is expressed with Sample.metadata: Any.
* SF\_SamplingFeature.parameter: NamedValue becomes Sample.parameter: NamedValue.

The **Table 9** summarizes the Sample mappings from the edition 2 Basic Samples package to edition 1.

**Table 9 — Sample mapping from 19156 edition 2 to edition 1 (informative)**

|  |  |  |
| --- | --- | --- |
| **Edition 2 class / property, Basic Samples package** | **Relation** | **Edition 1 class / property** |
| Sample | equivalent class | SF\_SamplingFeature |
| Sample.sampledFeature | equivalent property | SF\_SamplingFeature.sampledFeature |
| Sample.relatedObservation | equivalent property | SF\_SamplingFeature.relatedObservation |
| Sample.relatedSample | equivalent property | SF\_SamplingFeature.relatedSamplingFeature |
| Sample.metadata | has subProperty | SF\_SamplingFeature.lineage |
| Sample.parameter | equivalent property | SF\_SamplingFeature.parameter |
| Sample.sampling | has subProperty | SF\_Specimen.samplingMethod, SF\_Specimen.samplingTime, SF\_Specimen.samplingLocation |
| Sample.preparationStep | equivalent property | SF\_Specimen.processingDetails |
| SamplingProcedure | sub-class of | SF\_Process |
| PreparationProcedure | sub-class of | SF\_Process |
| Sampling |  | (no match) |
| Sampler |  | (no match) |

* + 1. Migration from SF\_SpatialSamplingFeature to SpatialSample

An instance of SF\_SpatialSamplingFeature class of Edition 1 can be expressed as an instance of the SpatialSample class of the Basic Samples package as follows (inherited properties of the SF\_SamplingFeature provided above not repeated here):

* SF\_SpatialSamplingFeature.hostedProcedure: OM\_Process becomes the Observer.observingProcedure: ObservingProcedure (observing procedures no longer associated with sampling features).
* SF\_SpatialSamplingFeature.positionalAccuracy: DQ\_PositionalAccuracy becomes a combination of SpatialSample.horizontalPositionalAccuracy: Any and SpatialSample.verticalPositionalAccuracy: Any.

For information about transitioning the specialized Spatial Sampling Feature types SF\_SamplingPoint, SF\_SamplingCurve, SF\_SamplingSurface and SF\_SamplingSolid of Edition 1 see the "Hard-typing vs. soft typing and codelist use" section below.

The **Table 10** summarizes the SpatialSample mappings from the edition 2 Basic Samples package to edition 1, including the properties inherited from the Sample and SF\_SamplingFeature.

**Table 10 — SpatialSample mapping from 19156 edition 2 to edition 1 (informative)**

|  |  |  |
| --- | --- | --- |
| **Edition 2 class / property, Basic Samples package** | **Relation** | **Edition 1 class / property** |
| SpatialSample | equivalent class | SF\_SpatialSamplingFeature |
| SpatialSample.sampledFeature | equivalent property | SF\_SpatialSamplingFeature.sampledFeature |
| SpatialSample.relatedObservation | equivalent property | SF\_SpatialSamplingFeature.relatedObservation |
| SpatialSample.relatedSample | equivalent property | SF\_SpatialSamplingFeature.relatedSamplingFeature |
| SpatialSample.metadata | has subProperty | SF\_SpatialSamplingFeature.lineage |
| SpatialSample.parameter | equivalent property | SF\_SamplingFeature.parameter |
| SpatialSample.sampling | has subProperty | SF\_Specimen.samplingMethod, SF\_Specimen.samplingTime, SF\_Specimen.samplingLocation |
| SpatialSample.preparationStep | equivalent property | SF\_Specimen.processingDetails |
| SpatialSample.shape | equivalent property | SF\_SamplingPoint.shape, SF\_SamplingCurve.shape, SF\_SamplingSurface.shape, SF\_SamplingSolid.shape |
| SpatialSample.horizontalPositionalAccuracy | sub-property of | SF\_SpatialSamplingFeature.positionalAccuracy |
| SpatialSample.verticalPositionalAccuracy | sub-property of | SF\_SpatialSamplingFeature.positionalAccuracy |

* + 1. Migration from SF\_Specimen to MaterialSample

An instance of SF\_Specimen class of Edition 1 can be expressed as an instance of the MaterialSample class of the Basic Samples package as follows (inherited properties of the SF\_SamplingFeature provided above not repeated here):

* SF\_Specimen.processingDetails: SF\_Process becomes MaterialSample.preparationStep: Conceptual Sample schema: PreparationStep and its processingDetails: Conceptual Sample schema: PreparationProcedure association. The attributes of the association class PreparationStep are mapped as follows:
  + PreparationStep.processOperator: CI\_ResponsibleParty is expressed as the metadata: Any association of the AbstractPreparationStep class in the Abstract Sample core package or any or another domain-specific realization of the PreparationStep interface.
  + PreparationStep.time: TM\_Object becomes AbstractPreparationStep.time: TM\_Object.
* SF\_Specimen.currentLocation: Location becomes MaterialSample.storageLocation: NamedLocation.
* SF\_Specimen.materialClass: GenericName is expressed using the AbstractSample.sampleType with appropriate code list values for sample material classification.
* SF\_Specimen.samplingLocation: GM\_Object becomes MaterialSample.sourceLocation: Geometry and/or Sampling.samplingLocation: Geometry via the MaterialSample.sampling association.
* SF\_Specimen.samplingMethod: SF\_Process becomes the Sampling.samplingProcedure: Conceptual Sample schema: SamplingProcedure via the MaterialSample.sampling association.
* SF\_Specimen.samplingTime: TM\_Object becomes Sampling.time: TM\_Object via the MaterialSample.sampling association.
* SF\_Specimen.size: Measure becomes MaterialSample.size: PhysicalDimension (multiple named size qualifiers possible with a dimenation and value: Measure for each).
* SF\_Specimen.specimenType: GenericName becomes AbstractSample.sampleType with appropriate code list values for domain specific sample classification (several sample types allowed).

The **Table 11** summarizes the MaterialSample mappings from the edition 2 Basic Samples package to edition 1, including the properties inherited from the Sample and SF\_SamplingFeature.

**Table 11 — MaterialSample mapping from 19156 edition 2 to edition 1 (informative)**

|  |  |  |
| --- | --- | --- |
| **Edition 2 class / property, Basic Samples package** | **Relation** | **Edition 1 class / property** |
| MaterialSample | equivalent class | SF\_Specimen |
| MaterialSample.sampledFeature | equivalent property | SF\_Specimen.sampledFeature |
| MaterialSample.relatedObservation | equivalent property | SF\_Specimen.relatedObservation |
| MaterialSample.relatedSample | equivalent property | SF\_Specimen.relatedSamplingFeature |
| MaterialSample.metadata | has subProperty | SF\_Specimen.lineage |
| MaterialSample.parameter | equivalent property | SF\_Specimen.parameter |
| MaterialSample.sampling | has subProperty | SF\_Specimen.samplingMethod, SF\_Specimen.samplingTime, SF\_Specimen.samplingLocation |
| MaterialSample.preparationStep | equivalent property | SF\_Specimen.processingDetails |
| MaterialSample.size | equivalent property | SF\_Specimen.size |
| MaterialSample.storageLocation | equivalent property | SF\_Specimen.currentLocation |
| MaterialSample.sourceLocation | equivalent property | SF\_Specimen.samplingLocation |

* 1. Observation and Sample collections

ISO 19156 Edition 1 did not include a concept of an Observation collection. In Edition 2 it is added as class ObservationCollection in package Basic Observations with the following attributes, associations and cardinalities:

* member: Conceptual Observation schema: Observation [0..\*]
* memberCharacteristics: ObservationCharacteristics [0..1]
* collectionType: AbstractObservationCollectionType [0..\*]
* relatedCollection: ObservationCollection [0..\*]
* metadata: Any [0..\*]

One concrete specialization of the AbstractObservationCollectionType class is provided in the Basic Observations package: ObservationCollectionType[[1]](#footnote-1) with an initial set of two values: ‘homogeneous’ and ‘summarizing’, which define how the properties of the ObservationCharacteristics instances associated with the ObservationCollection instance relate to the corresponding properties of the collection members (see clause Attribute collectionType). Other Observation collection classifications may be added by specializing the AbstractObservationCollectionType as required.

Edition 1 provided a collection of Sampling features as SF\_SamplingFeatureCollection class with a single association role member: SF\_SamplingFeature. In Edition 2 this is modelled as SampleCollection class in the Basic Samples package with the following attributes, associations and cardinalities:

* member: Conceptual Sample schema: Sample [0..\*]
* relatedCollection: SampleCollection [0..\*]
* metadata: Any [0..\*]

Unlike ObservationCollections, the SampleCollections are not classified, and do not have a dedicated mechanism for providing shared or summarized property values.

The **Table 12** summarizes the SampleCollection mappings from the edition 2 Basic Samples package to edition 1.

**Table 12 — SampleCollection mapping from 19156 edition 2 to edition 1 (informative)**

|  |  |  |
| --- | --- | --- |
| **Edition 2 class / property, Basic Samples package** | **Relation** | **Edition 1 class / property** |
| SampleCollection | equivalent class | SF\_SamplingFeatureCollection |
| SampleCollection.member | equivalent property | SF\_samplingFeatureCollection.member |
| SampleCollection.relatedCollection |  | (no match) |

* 1. Hard-typing vs. soft typing and codelist use

Observation classification by result type and SpatialSamplingFeature by the shape geometry type provided as sub-classes in the ISO 19156 Edition 1 are modelled using soft-typing based classification schemes in Edition 2 (AbstractObservationCharacteristics.observationType and AbstractSample.sampleType). This transition from hard-typing to soft-typing has been done to allow the use of the most appropriate Observation and Sample classification schemes to be used in the domain models, as well as to allow a single Observation and Sample instance to be classified using multiple classification schemes.

Concrete codelists are provided for both the result type based Observation classification and the shape geometry SpatialSample classification. Any additional Observation and Sample classification schemes can be provided in the domain models by extending the AbstractObservationType and AbstractSampleType classes, as illustrated for classification of Samples in Figure 33 below.



Figure 33 ­– (Example) Mechanism for defining a classification scheme for Samples based on the type of the sample material by extending the AbstractSampleType codelist.

Only an abstract, empty codelist extension point is provided for classifying Samplers. A wide variety of device types and methodologies used for creating samples are used in various domains, and any of these can be adopted in particular domain models by extending the AbstractSamplerType class. An example of this mechanism is illustrated as Figure 34.



Figure 34 – (Example) Mechanism for defining a generic classification scheme for Samplers by extending the AbstractSamplerType codelist.

* + 1. Migration of result type based Observation types

Instances of the specialized Observation types of 19156:2011 can be migrated into instances of the 19156:2020 Observation class of the Basic Observations package by providing an entry of the ObservationTypeByResultType codelist as a value of the observationType attribute as follows (labels provided here for readability, the corresponding URIs for the codelist entries should be used as specified in the code list vocabulary[[2]](#footnote-2)):

* OM\_Observation: Observation
* OM\_Measurement: Measurement
* OM\_CategoryObservation: Category Observation
* OM\_CountObservation: Count Observation
* OM\_TruthObservation: Truth Observation
* OM\_TemporalObservation: Temporal Observation
* OM\_GeometryObservation: Geometry Observation
* OM\_ComplexObservation: Complex Observation
* OM\_DiscreteCoverageObservation: Discrete CoverageObservation
* OM\_PointCoverageObservation: Point Coverage Observation
* OM\_TimeSeriesObservation: Time Series Observation
  + 1. Migration of geometry based sampling feature types

Instances of the specialized sampling feature types of 19156:2011 can be migrated into instances of the 19156:2020 SpatialSample class of the Basic Samples package by providing an entry of the SampleTypeByGeometryType codelist as a value of the sampleType attribute as follows (labels provided here for readability, the corresponding URIs for the entries should be used as specified in the code list vocabulary[[3]](#footnote-3)):

* SF\_SamplingPoint: Point Sample
* SF\_SamplingCurve: Curve Sample
* SF\_SamplingSurface: Surface Sample
* SF\_SamplingSolid: Solid Sample
  1. Generic metadata associations

In Edition 1 the Metadata association was provided only for the OM\_Observation class with type MD\_Metadata of ISO 19115:2003/Cor.1:2006 and with cardinality of 0..1. The ISO 19156 Edition 2 allows for providing metadata in addition to the concepts covered by the OMS model for most of the model classes:

* Abstract Observation core package:
  + AbstractObservationCharacteristics
  + AbstractObservingProcedure
  + AbstractObservableProperty
  + AbstractObserver
  + AbstractDeployment
  + AbstractHost
* Basic Observations package:
  + ObservationCollection
* Abstract Sample core package:
  + AbstractSample
  + AbstractSampling
  + AbstractSampler
  + AbstractPreparationStep
  + AbstractPreparationProcedure
  + AbstractSamplingProcedure
* Basic Samples
  + SampleCollection

Each of these classes contain an attribute with role name metadata of type Any and with cardinality of 0..\*. ISO 19115 metadata records may still be used for providing Observation instance metadata, but it is no longer the only allowed metadata model. With this change the ISO 19115 is also no longer a normative reference of the ISO 19156 Edition 2.

* 1. Discarded concepts

The ISO 19156 Edition 1 contained two requirementsClass packages with classes used in the UML but not specific to the Observations and Sampling features:

* General Feature Instance package:
  + GFI\_DomainFeature
  + GFI\_Feature
* Temporal Coverage package:
  + CVT\_DiscreteTimeInstantCoverage
  + CVT\_TimeInstantValuePair

The General Feature Instance package and its contained classes are not included in the Edition 2, as the General feature instances are no longer required in either the Observation or Sample models.

The Temporal Coverage package and its contained classes are not included in the Edition 2, as defining temporal coverages and characteristics of Observations with timeseries result values are considered out-of-scope for this specification. It is expected that the OGC Standard Timeseries Profile of Observations and Measurements (OGC 15-043r3) based on the 19156:2011 (Edition 1) UML model will be revised to profile the Edition 2 model instead, and to provide a detailed conceptual model for Observations with temporal coverage type results.

1. (informative)  
     
   Best practices in use of the Observation and Sampling models
   1. Features, coverages and observations — Different views of information

ISO 19109 describes the feature as a “fundamental unit of geographic information”. The “General Feature Model” (GFM) presented in ISO 19101 [21] and ISO 19109 defines a feature type in terms of its characteristic set of properties, including attributes, association roles, and behaviours, as well as generalization and specialization relationships, and constraints.

Typical concrete feature types have names like “road”, “watercourse”, “mine”, “atmosphere”, etc. For a road, the set of properties might include its name, its classification, the curve describing its centreline, the number of lanes, the surface material, etc. The complete description of a road instance, therefore, is the set of values for the set of properties that define a road type. This use of the feature model is object-centric, and supports a viewpoint of the world in terms of the set of discrete identifiable objects that occupy it.

The principal alternative model for geographic information is the coverage, described in ISO/DIS 19123-1 [26] [27]. This viewpoint focuses on the variation of a property within the (spatiotemporal) domain of interest. The domain might be a scene, a grid, a transportation network, a volume, a set of sampling stations, etc. The range of the coverage can be any property, such as reflectance, material type, concentration of some pollutant, number of lanes, etc. But the key to the coverage viewpoint is that it is property-centric, concerning the distribution of the values of a property within its domain space.

These viewpoints are not exclusive, and both are used in analysis and modelling. For example, a feature might be detected from the analysis of variation of a property in a region of interest (e.g. an ore-body from a distribution of assay values). Also, for some feature types, the value of one or more properties might vary across the feature, in which case the shape of the feature provides the coverage domain (e.g. ore-grade within a mine).

Observations focus on the data collection event. An act of Observation serves to assign a value to a property of a feature. If the property is non-constant, the value is a function or coverage. The results of a set of observations of different properties on the same feature-of-interest can provide a complete description of the feature instance. Alternatively, the results of a set of observations of the same property on a set of different features provide a discrete coverage of that property over a domain composed of the geometry of the feature set. The result of an observation of one property on one feature over time is a Temporal Coverage/Time-Series. The other properties of the Observation are metadata concerning the estimation of the value(s) of a property on a feature-of-interest.

In particular, Observations concern properties (e.g. shape, colour) whose values are determined using an identifiable procedure, in which there is a finite uncertainty in the result. This can be contrasted with properties whose values are specified by assertion (e.g. name, owner) and are therefore exact. The observation instance provides “metadata” for the property value-estimation process.

An observation event is clearly a “feature” in its own right, according to the GFM definition. An observation is an identifiable, instantiable and useful unit of information. Therefore, an observation is a feature type.

Transformation between viewpoints is frequently required.

This is illustrated in **Figure** 35, which schematically shows a dataset comprising values of a set of properties at a set of locations. A row of the table provides the complete description of the properties at a single location. This is a representation of a potential feature description. A column of the table describes the variation of a single property across the set of locations. This is a representation of a discrete coverage. A single cell in the table provides the value of a single property on a single feature. This might be the result of an observation.

Observations, Coverage and Feature representations might be associated with different phases of the data-processing cycle or value-chain:

* The observation view is associated with data collection, when an observation event causes values for a property of a feature to be determined, and during data entry when the data-store is updated by inserting values into fields in the datastore.
* A coverage view can be assembled from results of observations of a specific property, and represents data assembled for analysis, when the objective is to find signals in the variation of a property over a domain.
* A discrete feature description is a “summary” viewpoint, assembled from results of observation on the same target, or an “inferred” viewpoint, by extraction of a signal from a coverage.

Observations, Coverage and Feature representations are also often interlinked. Just as an Observation references the Feature it provides property information for, the Feature representation may also reference known Observations with more detailed property information. The same applies to Observations and Coverages; just as a Coverage can be the result of an Observation, an Observation can also be utilized to provide valuable meta-information on how the values being provided in the Range of the Coverage were derived.

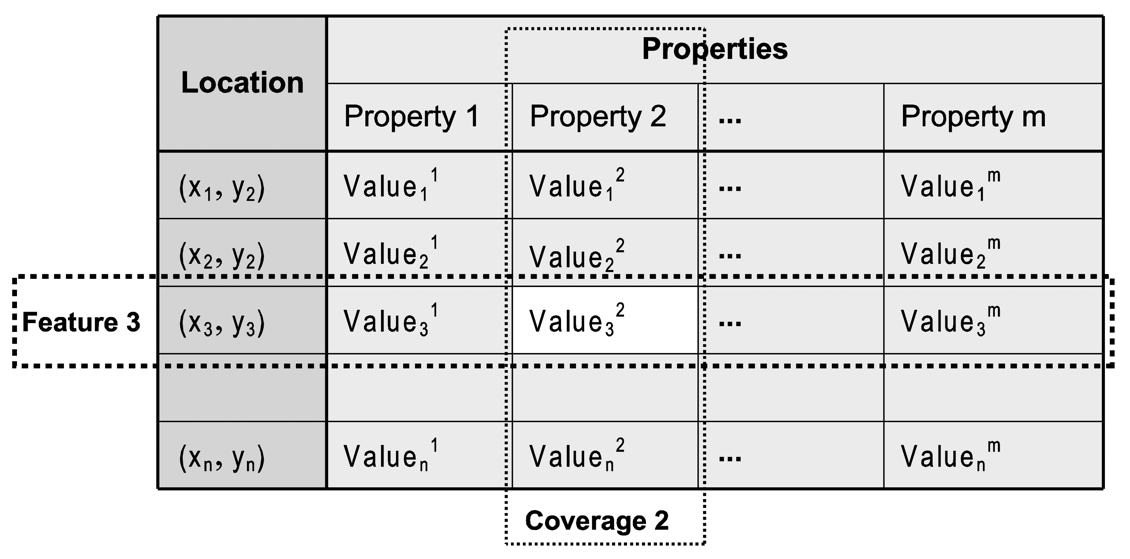


Figure 35 — Tabular representation of information associated with a set of locations

* 1. Observation concerns
     1. Domain specialization

Specialization of the observation model for an application domain is accomplished primarily using a domain model and its feature-type catalogue. For example, an instance of a feature type in the domain feature-type catalogue will provide the ultimate feature-of-interest for the investigation of which the observation is a part, and the characteristic properties of the feature type provide potential observed properties. A description of a sensor type or process familiar within the application domain is the value of the observation procedure, while the explicit device or person performing this procedure is provided as the observer.

The observation model encourages encapsulation of domain specialization in the associated classes, while the observation class itself rarely needs specialization. Nevertheless, other choices could be made in partitioning information between the classes in the model. For some applications, it might be convenient for information that is strictly associated with a second-layer object (procedure, feature-of-interest) to be associated with a specialized observation type.

For example, when measuring chemistry or contamination, the process often involves retrieving material samples from a sampling site, which are then sent to a laboratory for analysis. The material sample is a very tangible feature instance, with an identity. For some applications, it might be important to recognize the existence of the material sample, and retain a separate description of it. However, in other applications, particularly when the focus is on monitoring the change in a property at a sampling site, the existence of a series of distinct material samples is of minor or no interest. In this case, creating a series of objects and identifiers is superfluous to the user’s requirements.

In certain cases, some additional properties strictly associated with such a material sample must also be recorded, an example is the “sampling elevation” in a water or atmospheric column. A number of choices can be made. For example, the elevation could be:

1. a property of each distinct material sample on which atomic observations are actually made,
2. a property of the sampling site(which would require distinct sites for all elevations at which observations are made),
3. a parameter of the observation procedure (which makes the procedure specific to this observation series only), or
4. a parameter of the observation event, either using the soft-typed arbitrary event-specific parameter, or through specialization of the observation type.

Any of these is a legitimate approach. The optimum one will be dependent on the application.

All of the classes in the models presented here for observations and procedures can be further specialized for domain-specific purposes, whereby the abstract classes provided in the Abstract Core models have been specifically foreseen as a neutral basis for such domain extensions. Additional attributes and associations can be added as necessary.

EXAMPLE “Assay” might be derived from Observation, fixing the observedProperty to be “ChemicalConcentration” and adding an additional attribute “analyte”.

* + 1. Comparison with provider-oriented models

The OMS model is intended to provide a basic output- or user-oriented information model for sensor web and related applications. The goal is to provide a common language for discourse regarding sensor, sample and observation systems.

In comparison, SensorML [10] has a process- or provider-oriented data model. These are usually used to describe data at an early stage in the data processing and value-adding chain. This might be prior to the details of the feature-of-interest and observed property being assembled and assigned to the result in a way that carries the key semantics to end-users of observation data. In particular, part of a SensorML datastream might include information that must be processed to determine the position of the target or feature-of-interest. At the early processing stage such positional and timing information might be embedded within the result.

Nevertheless, even within these low-level models the OMS formalization can be applied. The proximate feature-of-interest is the vicinity of the sensor. The observed property is a composite type including components representing observation timing, and position and attitude of a sensor, etc. This must be processed to obtain the details of the ultimate feature-of-interest. The procedure is a description of sensor methodology including elements that capture all of the elements of the composite characteristic or property type, etc. while the observer references the explicit sensor utilized.

* + 1. Observation discovery and use

The OMS model presented here offers a user-oriented viewpoint. The information object is characterized by a small set of properties, which are likely to be of interest to a user for discovery and request of observation data. The user will typically be interested primarily in a feature-of-interest, or the variation of a characteristic. The model provides these items as first-order elements. An interface to observation information should expose these properties explicitly.

Observation discovery and use is often done querying APIs; although with LinkedData practices being more and more used, one might discover an observation simply because an instance of a domain feature uses its URI or it has been crawled by a search engine bot.

Observation oriented APIs, be them from the previous generation (OGC SOS [11] ) or the current one (OGC SensorThings API[12]) share commonalities in the way they approach this topic. They both leverage the OMS model to directly allow filtering on featureOfInterest, observedProperty and procedure.

* The SensorThings API model and OData query graph allow filtering on all aspects of the observational data model, both for discovery and data retrieval (both ‘operations’ being intertwined in the REST pattern)
* SOS [11], having these three concepts as classifiers for an observationOffering in the capabilities description, allows them to be used for discovery and as explicit parameters in the GetObservation request

From a user point of view, these associated objects (procedure, target feature, characteristic) are primarily metadata, which are only of interest to specialists during discovery, and then to assist evaluation or processing of individual results.

Each of these associated objects might require a complex description. Hence they are modelled as distinct classes, which can be as simple or complex as necessary.

In a serialized representation (e.g. JSON, XML following the GML pattern, etc…), they might appear inline, perhaps described using one of the models presented here, or they can be indicated by reference using a URI [3]. The URI identifier might be a URL link or service call, which should resolve immediately to yield a complete resource. Or it might be a canonical identifier, such as a URN, which the user and provider are preconfigured to recognize and understand.

On the other hand, SensorML takes a process- or provider-oriented viewpoint. Discovery and request is based primarily on the user having knowledge of specific sensor systems and their application. While this is a reasonable assumption within technical communities, specialist knowledge of sensor systems would not be routinely available within a broader set of potential users of sensor data, particularly as this is made widely available through interfaces like OGC SensorThings API and SOS.

* + 1. Observations, interpretations, simulations

Some conceptual frameworks make a fundamental distinction between observations, interpretations and simulations as the basis for their information modelling approach. This supports a pattern in which observations are given precedence and archived, while interpretations or simulations are more transient, being the result of applying the current algorithms and paradigms to the currently available observations.

An alternative view is that the distinction is not absolute, but is one of degree. Even the most trivial "observations" are mediated by some theory or procedure. For example, the primary measurement when using a mercury-in-glass thermometer is the position of the meniscus relative to graduations. This allows the length of the column to be estimated. A theory of thermal expansion plus a calibration for the physical realization of the instrument allows conversion to an inferred temperature. Other observations and measurements all involve some kind of processing from the primary observable property. For modern instruments, the primary observable property is almost always voltage or resistance or frequency from some kind of sensing element, so the "procedure" typically involves calibrations, etc., built on a theory of operation for the sensor. Pertaining to simulations, the OMS model allows for the description of simulated observations (ex : forecast) and can capture entire processing chains starting from initial observation(s) (e.g. surface/ground water level, rainfall) to generate corresponding forecasts scenarios (e.g. flood, drought) through the use of simulation algorithms. Similarly, aggregates can be calculated (e.g. averages over space or time) and provided referencing their primary source observations.

However, the same high-level information model — that every "value" is an estimate of the value of a property, generated using a procedure and inputs — applies to "observations", "interpretations" as well as “simulations”. It is just that the higher the semantic value of the estimate, the more theory and processing is involved.

Within the model provided in this standard, there is no conceptual distinction between observations, interpretations and simulations. The OMS model allows for the description of the observational workflow together with the explicit description of the processing chain instance that has taken a more primitive observation (e.g. an image) and retrieved a higher level observation (e.g. the presence of a certain type of feature instance) through the application of one or more processing steps. The result is the entire continuum of primary and processed data provided in a harmonized model.

* 1. Sample, Sampling concerns
     1. Sample as observation-collector

The sample model provides

1. an intermediate Sample class that allows the assignment of primitive and intermediate properties within a processing chain,
2. three sub-types of Samples corresponding to practices applied by communities where Sample are either defined by their geospatial characteristics, statistical characteristics or their material ones (being taken ex-situ for further observation), and
3. additional classes providing a context for the description of sampling acts and regimes.
4. In addition, the sample model allows for references to observation(s) concerning a shared common feature-of-interest / sampledFeature. This provides an access route to observation information that is convenient under some project scenarios, where the sampling strategy provides the logical organization of observations.

EXAMPLE An observational mission or campaign might organize its data according to flightlines, ship's tracks, outcrops, sampling-stations, quadrats, etc., or an observation archive or museum might organize observations by specimen (a specific type of material sample targeting preservation).

* + 1. Observation feature(s)-of-interest

Application of the OMS model requires careful attention to identify the feature-of-interest context correctly. This can be straightforward if the observation is clearly concerned with an easily identified concrete feature type from a domain model. However, the ultimate feature-of-interest to the investigator might not be the proximate feature-of-interest for the observation. In some cases, a careful analysis reveals that the type of the feature-of-interest had not previously been identified in the application domain.

The key is that the proximate feature-of-interest must be capable of carrying this result as the value or component of the value of a relevant property. So a useful approach in analysis is to consider what the result of the observation is, and then the feature-of-interest can be deduced since it must have a property with this result as its value. If an observation produces a result with several elements, or if there are a series of related observations with different results, then this might help further refine the understanding of the type of the ultimate feature-of-interest.

EXAMPLES

* In groundwater monitoring the ultimate feature-of-interest is often a given hydrogeological unit but the proximate feature-of-interest is the Well (or a more precise Feature) where the Observation occurs.
* In air quality monitoring the ultimate feature-of-interest is either the general atmosphere or alternatively a defined region (e.g. air quality zone) the monitoring facility, while the proximate feature-of-interest is the bubble of air around the air intake of the monitoring facility
* In surface water quality monitoring the ultimate feature-of-interest is a river (or a section of it) but the proximate feature-of-interest of the observation is a vial of water (material sample) on which analysis are conducted in a laboratory.
  + 1. Processing chains and intermediate features-of-interest

The Observation model implies a direct relationship between the observed property and the type of the feature-of-interest (e.g. a material sample type has a property ‘mass’ and the observation’s observed property is ‘mass’). However, as discussed in section 7.2.2.2 the relationship between the observed property and property(ies) of the ultimate feature-of-interest is often more complex.

The Sample model is a mechanism for preserving the strict association, by providing a specific intermediate feature type whose observable properties are unspecified in advance, but supplied through an unlimited set of related observations. The path from a sensed property obtained through observations related to the sample, to the interesting property on the ultimate feature-of-interest, is modelled as a processing chain.

If intermediate values are explicit, then the processing chain can be modelled as a sequence of “observations”, with intermediate features of interest carrying intermediate property types. Each intermediate value must apply to a feature-of-interest that bears this property, or a sampling feature. Note that the types of these features might not be conventional or immediately recognisable, but the coherence of the OMS model does imply their existence. Hence, if any intermediate result is made explicit, then a suitable intermediate feature must also be identified.

* 1. Observations and Coverages

Within the Open Geospatial Consortium (OGC), different data models have evolved for the provision of sensor data (Observations, measurements and samples Model (OMS)) and datacubes (OGC Coverage Implementation Schema (ISO 19123-2:2018 & ISO/DIS 19123-1) (CIS))[16]. While these models are formally distinct, and were developed mostly independently of each other, there are great similarities as well as overlaps between these models. At its core, the OMS model provides an exact description of how an observation or measurement value came to be, while the CIS model has concerned itself with providing alignment with a spatial swath and data recorded for this region; these differing approaches have led to a focus on different aspects of the entirety of observational data and measurement metadata. In addition, these models are often used conjunctively, as each model provides relevant information missing from the other model.

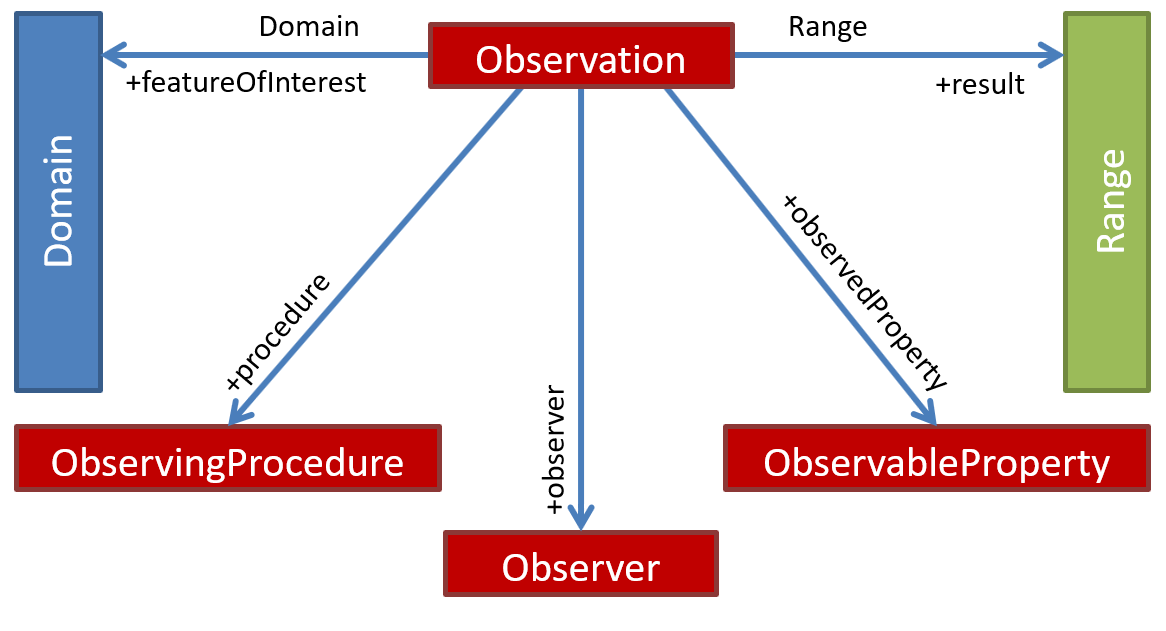


Figure 36 — OMS model key elements

Upon closer analysis it becomes clear just how similar these models are at their core, as well as how each provides concepts essential for precise description of the data that have been neglected within the other models. Both OMS and CIS provide a set of values (Range) over a given extent (Domain). However, while the CIS model provides information on the explicit points within the Domain extent for which values are provided (domainSet, usually some sort of grid) as well as the mapping of these points to these values provided within the Range (provided via the coverageFunction), the OMS model provides far more detailed information on the measurement methodology and process via the ObservableProperty, ObservingProcedure and Observer types.

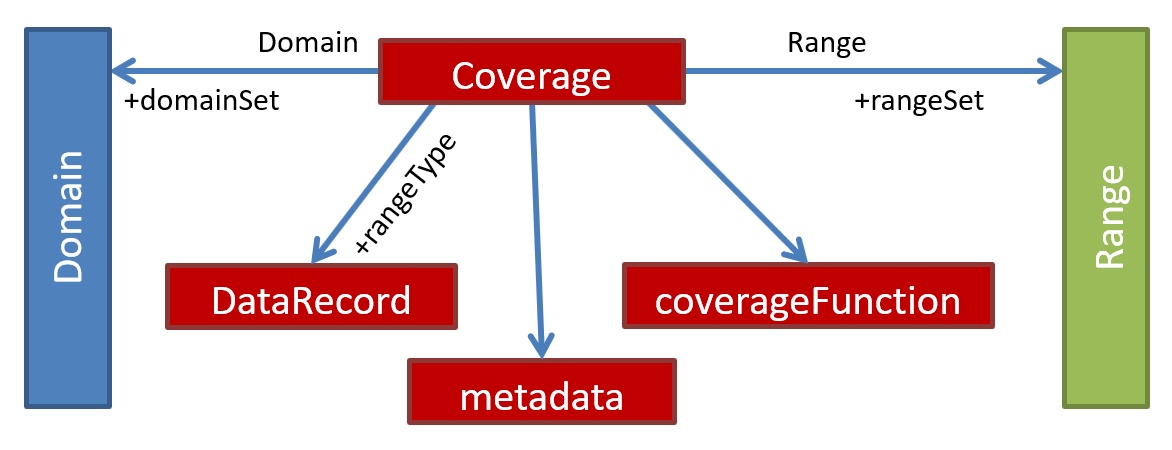


Figure 37 — CIS model key elements.

When OMS and CIS models are used in conjunction, care must be taken in ensuring alignment pertaining to the Domains being referenced. The observation community often provides domain features with a bounding polygon as the Domain of complex Observations, assuming the explicit points within for which values are provided to be contained within the coverage provided as a result of the Observation. Under the CIS model the Domain is always provided together with some explicit representation of the actual points within the Domain for which values are provided, e.g. origin and offsets for the definition of regular grid points. For example, when providing data on a transect or vertical profile, the ultimateFeatureOfInterest (OMS Domain) may reference a feature representing this transect or profile, while the Coverage provided as result (OMS Range) contains both the explicit measurement locations (CIS Domain), often as offsets along the given transect or profile, and the measurement values (CIS Range).

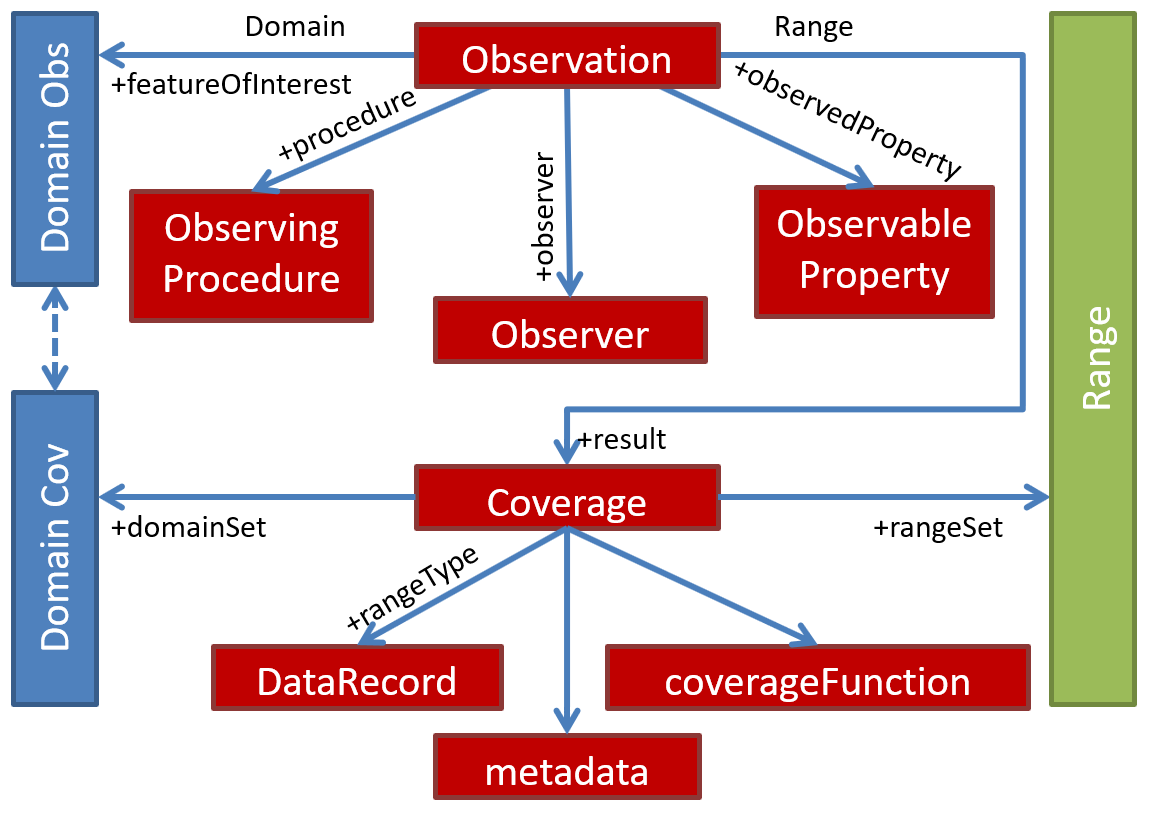


Figure 38 — Coverage as a result of an Observation.

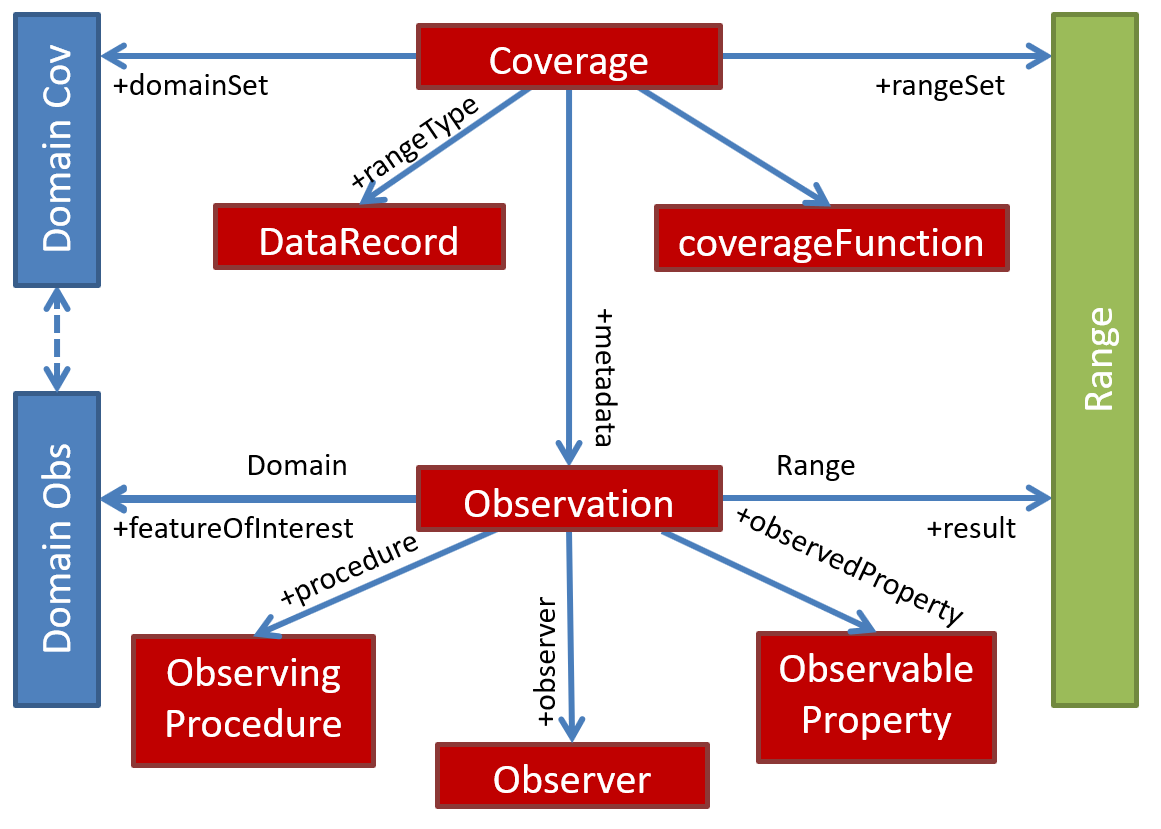


Figure 39 — Observation as metadata of a Coverage.

Conversely to the model described above, OMS Observations have long been utilized for the provision of more explicit metadata on how the values provided in the rangeSet have been ascertained, whereby the Observation result was left as void. In this updated version, the ObservationCharacteristics type has been foreseen for utilization or extension within this context, as the constraints on this type are far looser than on the Observation. When OMS and CIS models are used in conjunction, it is recommended that the OMS Domain provided as ultimateFeatureOfInterest is an envelope of the CIS Domain.

Bibliography

1. Chrisman, N.R. *Exploring Geographical Information Systems*, 2nd Edition. Wiley. 2001
2. Fowler, M. *Analysis Patterns: reusable object models*. Addison Wesley Longman, Menlo Park, CA. 1998
3. IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax.* August 1998
4. *VIM3: International vocabulary of metrology – Basic and general concepts and associated terms* : BIPM/ISO 2012
5. Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A. (1971), *Foundations of measurement, Vol. I: Additive and polynomial representations*, New York: Academic Press
6. Luce, R.D., Krantz, D.H., Suppes, P., Tversky, A. (1990), *Foundations of measurement, Vol. III: Representation, axiomatization, and invariance*, New York: Academic Press
7. Nieva, T. *Remote data acquisition of embedded systems using internet technologies: a role-based generic system specification.* Thesis, Ecole Polytech. Fed. Lausanne 2001. Available (viewed 2020-09-29) at <http://infoscience.epfl.ch/record/313/files/Nieva01.pdf>
8. Sarle, W.S., *Measurement theory: frequently asked questions*. Originally published in the Disseminations of the International Statistical Applications Institute, 4th edition, 1995, Wichita: ACG Press, pp. 6166. Revised 1996, 1997. Available (viewed 2021-10-21) at <https://www.academia.edu/3337298/Measurement_theory_Frequently_asked_questions>
9. Schadow, G., McDonald, C.J. (eds.), *UCUM, Unified Code for Units of Measure*. Available (viewed 2020-09-29) at https://ucum.org/ucum.html. Tentative ontology at <http://finto.fi/ucum/en/> (viewed 2020-09-24)
10. *Sensor Model Language (SensorML)*, OpenGIS® Implementation Standard, OGC 12-000r2. Available (viewed viewed 2020-09-29) at <http://www.opengeospatial.org/standards/sensorml>
11. *Sensor Observation Service*, OpenGIS® Implementation Specification OGC document 12-006
12. The OGC SensorThings API Part 1: Sensing (2016). OGC Document OGC: 15-078R6,
13. Suppes, P., Krantz, D.H., Luce, R.D., Tversky, A. (1989), *Foundations of measurement, Vol. II: Geometrical, threshold, and probabilistic representations*, New York: Academic Press
14. *SWE Common Data Model Encoding Standard,* OpenGIS® Implementation Standard OGC document 08094r1
15. *OGC: The Specification Model - A Standard for Modular specifications (2009). OGC document 08-131r3,*
16. K. Schleidt and P. Baumann, "Interconnecting Sensor Data and Datacubes," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, 2019, pp. 5555-5558, doi: 10.1109/IGARSS.2019.8898232.
17. *QUDT - Quantities, Units, Dimensions and Data Types Ontologies*. Ralph Hodgson; Paul J. Keller; Jack Hodges; Jack Spivak. Available (viewed 2020-09-29) at <http://www.qudt.org/>
18. *Semantic Sensor Network Ontology.*  Armin Haller, Krzysztof Janowicz, Simon Cox, Danh Le Phuoc, Kerry Taylor, Maxime Lefrançois. Available (viewed 2020-09-29) at <https://www.w3.org/TR/vocab-ssn/>
19. *Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE*. Sylvain Grellet , Gerhard Dünnebeil, Anders Foureaux, Carsten Hollmann, Frédéric Houbie, Diomede Illuzzi, Simon Jirka, Barbara Magagna, Matthes Rieke, Alessandro Sarretta, Katharina Schleidt, Paweł Soczewski, Paolo Tagliolato, Mickael Treguer and Alexander Kotsev, Michael Lutz. Available (viewed 2020-09-29) at <https://inspire.ec.europa.eu/id/document/tg/d2.9-o%26m-swe>
20. *Ontology for observations and sampling features, with alignments to existing models*. S.J.D. Cox. Semantic Web. 2017. Available (viewed 2020-09-29) at https://content.iospress.com/articles/semantic-web/sw214
21. ISO 19101-1:2014 *Geographic information — Reference model — Part 1: Fundamentals*
22. ISO 19105:2000 *Geographic information — Conformance and testing*
23. ISO/DIS 19105 *Geographic information — Conformance and testing*
24. ISO 19115-1:2014 *Geographic information — Metadata — Part 1: Fundamentals*
25. ISO 19115-1:2014/Amd 2:2020 *Geographic information — Metadata — Part 1: Fundamentals — Amendment 2*
26. ISO/DIS 19123-1 *Geographic information — Schema for coverage geometry and functions — Part 1 Fundamentals*
27. ISO 19123-2:2018 *Geographic information — Schema for coverage geometry and functions — Part 2: Coverage implementation schema*
28. ISO 19136-1:2020 *Geographic information — Geography Markup Language (GML) — Part 1: Fundamentals*
29. ISO 19157:2013 *Geographic information — Data quality*
30. ISO 19157:2013/Amd 1:2018 *Geographic information — Data quality — Amendment 1: Describing data quality using coverages*
31. *ISO/DIS 19157-1 Geographic information — Data quality — Part 1: General requirements*

1. http://www.opengis.net/def/observation-collection-type/OGC/0/by-member-characteristics-semantics [↑](#footnote-ref-1)
2. http://www.opengis.net/def/observation-type/OGC/0/by-result-type [↑](#footnote-ref-2)
3. http://www.opengis.net/def/sample-type/OGC/0/by-geometry-type [↑](#footnote-ref-3)