

Open Geospatial Consortium

Submission Date: 2023-04-26

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: 20-082r4

Version: 3.0.0

Category: OGC® Abstract Specification

Editors: Katharina Schleidt, Ilkka Rinne

OGC Abstract Specification Topic 20: Observations, measurements and samples

Copyright notice

Copyright © 2023 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.

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.

Document type:	OGC® Abstract Specification
Document subtype:	
Document stage:	Draft
Document language:	English

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.

Note on the OGC and ISO TC 211 Standards alignment

This revised OGC Standard was created from ISO 19156:2023(E) document after the final edits by the ISO Central Secretariat for publication as an ISO International Standard. It has an identical content with ISO 19156:2023(E) except for the copyright, title pages and double branding.

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 Sörés	Hungarian Mining and Geological Service
Jörg Klausen	World Meteorological Organization (WMO)

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 19156:2023(E)

ISO TC 211/WG 9

Date: 2023-04

Geographic information — Observations, measurements and samples

© ISO 2023

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

<i>Foreword</i>	xv
<i>Introduction</i>	xvi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Document conventions	5
4.1 Abbreviated terms and acronyms	5
4.2 Schema language	6
4.3 Model element names	6
4.4 Requirements and recommendations	6
4.5 Requirements classes	7
4.6 Conformance classes	8
4.7 Identifiers	8
4.8 Associations in UML context diagrams	10
5 Conformance	10
5.1 Overview	10
5.2 Conformance classes	10
6 Packaging, requirements and dependencies	13
6.1 Requirements	13
6.2 UML	14
6.2.1 UML package structure.....	14
6.2.2 UML package dependencies	14
6.3 Note on the use of "Any"	16
7 Fundamental characteristics of observations and samples (informative)	17
7.1 Observation schema	17
7.1.1 Property evaluation	17
7.1.2 Observation	17
7.1.3 Properties of an Observation.....	18
7.1.4 Observation location.....	18
7.1.5 Result types.....	19
7.1.6 Use of the observation model	19
7.2 Sample schema	19
7.2.1 Role of sample features	19
7.2.2 Proximate vs. ultimate feature-of-interest	19
7.2.3 Role of samples.....	21
7.2.4 Sampling process	21
7.2.5 Classification of samples	22
7.3 Alignment between Observation, Sample and domain models	22
7.3.1 Model consistency	22

7.3.2	Relationship between Sample and domain features	25
8	<i>Conceptual Observation schema</i>	27
8.1	General	27
8.1.1	Conceptual Observation model.....	27
8.1.2	Conceptual Observation schema package Requirements Class	28
8.1.3	Association relatedObservation	29
8.2	Observation.....	29
8.2.1	Observation Requirements Class	29
8.2.2	Interface Observation	30
8.2.3	Attribute phenomenonTime	30
8.2.4	Attribute resultTime.....	31
8.2.5	Attribute validTime	31
8.2.6	Association featureOfInterest.....	31
8.2.7	Association observedProperty	32
8.2.8	Association result	32
8.2.9	Association observingProcedure	33
8.2.10	Association observer.....	33
8.2.11	Association host	33
8.2.12	Constraint Observer or Host	33
8.2.13	Constraint ObservableProperty characteristic associated with featureOfInterest.....	33
8.2.14	Constraint suitable ObservableProperty.....	33
8.2.15	Constraint suitable result type.....	34
8.2.16	Constraint unit of measure	34
8.3	ObservableProperty	34
8.3.1	ObservableProperty Requirements Class.....	34
8.3.2	Interface ObservableProperty	34
8.3.3	Association observer	35
8.4	Procedure.....	35
8.4.1	Procedure Requirements Class	35
8.4.2	Interface Procedure.....	35
8.5	ObservingProcedure	36
8.5.1	ObservingProcedure Requirements Class	36
8.5.2	Interface ObservingProcedure	36
8.5.3	Association observer	37
8.6	Observer	37
8.6.1	Observer Requirements Class	37
8.6.2	Interface Observer	37
8.6.3	Association observableProperty.....	38
8.6.4	Association observingProcedure	38
8.6.5	Association deployment.....	38
8.7	Host.....	38
8.7.1	Host Requirements Class.....	38
8.7.2	Interface Host.....	38
8.7.3	Association deployment.....	39
8.7.4	Association relatedHost	39
8.8	Deployment	39
8.8.1	Deployment Requirements Class	39
8.8.2	Interface Deployment	39
8.8.3	Association observer	40
8.8.4	Association host	40
9	<i>Abstract Observation Core</i>	40

9.1 General.....	40
9.1.1 Abstract Observation Core Package Requirements Class	40
9.1.2 Association metadata	41
9.2 AbstractObservationCharacteristics	41
9.2.1 AbstractObservationCharacteristics Requirements Class.....	41
9.2.2 Feature type AbstractObservationCharacteristics	43
9.2.3 Attribute observationType.....	44
9.2.4 Attribute parameter.....	44
9.2.5 Attribute resultQuality.....	45
9.2.6 Association proximateFeatureOfInterest	45
9.2.7 Association ultimateFeatureOfInterest	45
9.2.8 Association collection	46
9.3 AbstractObservation.....	46
9.3.1 AbstractObservation Requirements Class.....	46
9.3.2 Constraint observationType.....	47
9.3.3 Constraint resultTime instant	47
9.3.4 Constraint parameter unique name	48
9.3.5 Constraint proximate or ultimate featureOfInterest.....	48
9.3.6 Constraint Observer or Host	48
9.3.7 Constraint ObservableProperty characteristic associated with featureOfInterest.....	48
9.3.8 Constraint suitable ObservableProperty.....	48
9.3.9 Constraint suitable result type.....	48
9.4 AbstractObservableProperty.....	48
9.4.1 AbstractObservableProperty Requirements Class	48
9.5 AbstractObservingProcedure	49
9.5.1 AbstractObservingProcedure Requirements Class	49
9.6 AbstractObserver	51
9.6.1 AbstractObserver Requirements Class	51
9.7 AbstractHost	52
9.7.1 AbstractHost Requirements Class.....	52
9.8 AbstractDeployment.....	54
9.8.1 AbstractDeployment Requirements Class	54
9.8.2 Attribute deploymentReason	55
9.8.3 Attribute deploymentTime	56
9.9 AbstractObservationCollection	56
9.9.1 AbstractObservationCollection Requirements Class	56
9.9.2 Feature type AbstractObservationCollection	57
9.9.3 Attribute collectionType	57
9.9.4 Association member	58
9.9.5 Association memberCharacteristics.....	58
9.9.6 Association relatedCollection	58
9.10 NamedValue	58
9.10.1 NamedValue Requirements Class.....	58
9.10.2 Data type NamedValue.....	59
9.10.3 Attribute name	59
9.10.4 Attribute value.....	59
9.11 Codelists	59
9.11.1 AbstractObservationType	59
9.11.2 AbstractObservationCollectionType	59
10 Basic Observations	60

10.1 General	60
10.1.1 Basic Observations Package Requirements Class	60
10.1.2 Attribute link.....	61
10.1.3 Attribute location.....	61
10.2 Observation.....	61
10.2.1 Observation Requirements Class	61
10.3 ObservationCharacteristics.....	63
10.3.1 ObservationCharacteristics Requirements Class	63
10.4 ObservationCollection	63
10.4.1 ObservationCollection Requirements Class	63
10.5 ObservingCapability	63
10.5.1 ObservingCapability Requirements Class.....	63
10.5.2 Feature type ObservingCapability.....	66
10.6 ObservableProperty	67
10.6.1 ObservableProperty Requirements Class	67
10.7 ObservingProcedure	68
10.7.1 ObservingProcedure Requirements Class.....	68
10.8 Observer	70
10.8.1 Observer Requirements Class	70
10.9 Host.....	73
10.9.1 Host Requirements Class	73
10.10 Deployment.....	75
10.10.1 Deployment Requirements Class	75
10.11 GenericDomainFeature.....	77
10.11.1 GenericDomainFeature Requirements Class	77
10.11.2 Feature type GenericDomainFeature	79
10.12 Codelists	79
10.12.1 ObservationCollectionType.....	79
10.12.2 ObservationTypeByResultType	81
11 Conceptual Sample schema.....	83
11.1 General	83
11.1.1 Conceptual Sample schema model	83
11.1.2 Conceptual Sample Schema package Requirements Class	84
11.2 Sample	84
11.2.1 Sample Requirements Class	84
11.2.2 Interface Sample	85
11.2.3 Association sampling.....	85
11.2.4 Association preparationStep.....	85
11.2.5 Association sampledFeature	85
11.2.6 Association relatedSample.....	86
11.3 Sampling	86
11.3.1 Sampling Requirements Class	86
11.3.2 Interface Sampling	87
11.3.3 Association sample	87
11.3.4 Association featureOfInterest.....	87
11.3.5 Association sampler	87
11.3.6 Association samplingProcedure	88
11.3.7 Association relatedSampling.....	88

11.4 Sampler	88
11.4.1 Sampler Requirements Class	88
11.4.2 Interface Sampler	88
11.4.3 Association sampling	89
11.4.4 Association implementedProcedure	89
11.5 PreparationStep.....	89
11.5.1 PreparationStep Requirements Class	89
11.5.2 Interface PreparationStep	89
11.5.3 Association processingDetails	90
11.5.4 Association preparedSample.....	90
11.6 PreparationProcedure.....	90
11.6.1 PreparationProcedure Requirements Class.....	90
11.6.2 Interface PreparationProcedure.....	90
11.6.3 Association samplePreparationStep.....	90
11.7 SamplingProcedure.....	91
11.7.1 SamplingProcedure Requirements Class	91
11.7.2 Interface SamplingProcedure	91
11.7.3 Association sampling	91
11.7.4 Association sampler.....	91
12 Abstract Sample Core	92
12.1 General.....	92
12.1.1 Abstract Sample Core Package Requirements.....	92
12.2 AbstractSample.....	92
12.2.1 AbstractSample Requirements Class	92
12.2.2 Attribute sampleType	95
12.2.3 Attribute parameter	95
12.3 AbstractSampling.....	95
12.3.1 AbstractSampling Requirements Class	95
12.3.2 Attribute samplingLocation	96
12.3.3 Attribute time	97
12.3.4 Attribute parameter	97
12.4 AbstractSampler	97
12.4.1 AbstractSampler Requirements Class.....	97
12.4.2 Attribute samplerType.....	98
12.5 AbstractSamplingProcedure.....	99
12.5.1 AbstractSamplingProcedure Requirements Class	99
12.6 AbstractPreparationProcedure.....	100
12.6.1 AbstractPreparationProcedure Requirements Class	100
12.7 AbstractPreparationStep.....	101
12.7.1 AbstractPreparationStep Requirements Class.....	101
12.7.2 Attribute description	102
12.7.3 Attribute time	102
12.8 Codelists	102
12.8.1 AbstractSampleType.....	102
12.8.2 AbstractSamplerType	102
13 Basic Samples.....	103
13.1 General.....	103
13.1.1 Basic Samples Package Requirements Class	103
13.2 Sample.....	104

13.2.1	Sample Requirements Class	104
13.3	SpatialSample.....	106
13.3.1	SpatialSample Requirements Class	106
13.3.2	Feature type SpatialSample	106
13.3.3	Attribute shape	107
13.3.4	Attribute horizontalPositionalAccuracy	107
13.3.5	Attribute verticalPositionalAccuracy.....	107
13.4	MaterialSample	107
13.4.1	MaterialSample Requirements Class	107
13.4.2	Feature type MaterialSample	108
13.4.3	Attribute size.....	108
13.4.4	Attribute storageLocation	108
13.4.5	Attribute sourceLocation	108
13.5	StatisticalSample	109
13.5.1	StatisticalSample Requirements Class	109
13.5.2	Feature type StatisticalSample	109
13.5.3	Attribute classification	109
13.6	Sampling	110
13.6.1	Sampling Requirements Class	110
13.7	Sampler.....	111
13.7.1	Sampler Requirements Class.....	111
13.8	SamplingProcedure	113
13.8.1	SamplingProcedure Requirements Class	113
13.9	PreparationProcedure	114
13.9.1	PreparationProcedure Requirements Class	114
13.10	PreparationStep	116
13.10.1	PreparationStep Requirements Class.....	116
13.11	SampleCollection	117
13.11.1	SampleCollection Requirements Class.....	117
13.11.2	Feature type SampleCollection	118
13.11.3	Association member	119
13.11.4	Association relatedCollection	119
13.12	PhysicalDimension.....	119
13.12.1	PhysicalDimension Requirements Class.....	119
13.12.2	Data type PhysicalDimension.....	119
13.12.3	Attribute dimension.....	119
13.12.4	Attribute value	120
13.13	NamedLocation	120
13.13.1	NamedLocation Requirements Class	120
13.13.2	Data type NamedLocation	120
13.13.3	Attribute address	120
13.13.4	Attribute name.....	121
13.13.5	Attribute representativeGeometry	121
13.14	StatisticalClassification	121
13.14.1	StatisticalClassification Requirements Class	121
13.14.2	Data type StatisticalClassification	121
13.14.3	Attribute concept.....	121
13.14.4	Attribute classification	122
13.15	Codelists	122
13.15.1	SampleTypeByGeometryType.....	122

Annex A (normative) Abstract test suite.....	123
A.1 Abstract tests for Conceptual Observation schema package.....	123
A.2 Abstract tests for Abstract Observation Core package.....	124
A.3 Abstract tests for Basic Observations package.....	127
A.4 Abstract tests for Conceptual Sample schema package	129
A.5 Abstract tests for Abstract Sample Core package	130
A.6 Abstract tests for Basic Samples package	132
Annex B (informative) Common usage of OMS concepts	135
B.1 Introduction.....	135
B.2 Earth observations (EO)	135
B.3 Metrology.....	136
B.4 Earth science simulations	136
B.5 Assay/Chemistry.....	136
B.6 Geology field observations.....	137
B.7 Geotechnics observations	137
B.8 Water quality observations.....	138
B.9 Soil quality observations	139
Annex C (informative) Changes in the Observation and Sample models between ISO 19156:2011 and ISO 19156:2023 (this document)	140
C.1 General.....	140
C.2 Package and requirements class structure.....	140
C.3 Interfaces in the conceptual schema packages	142
C.4 Realizations of the conceptual schemas as abstract and concrete feature type classes.....	143
C.5 Modelling of the Observation concept	144
C.5.1 OM_Observation in ISO 19156:2011	144
C.5.2 Observation in ISO 19156:2023	144
C.5.3 Migration from OM_Observation to Observation	147
C.6 Modelling of the Sample and Sampling concepts	148
C.6.1 SF_SamplingFeature, SF_Specimen SF_SpatialSamplingFeature and in ISO 19156:2011	148
C.6.2 Sample, SpatialSample, MaterialSample and StatisticalSample in ISO 19156:2023	150
C.6.3 Modelling of environmental monitoring stations.....	151
C.6.4 Migration from SF_SamplingFeature to Sample.....	151
C.6.5 Migration from SF_SpatialSamplingFeature to SpatialSample	152
C.6.6 Migration from SF_Specimen to MaterialSample.....	153
C.7 Observation and Sample collections.....	155
C.8 Hard-typing vs. soft-typing and codelist use	156
C.8.1 Migration of result-type-based Observation types.....	157
C.8.2 Migration of geometry-based sampling feature types	158
C.9 Generic metadata associations	158
C.10 Discarded concepts	159

Annex D (informative) Best practices in use of the Observation and Sampling models	161
D.1 Features, Coverages and Observations — Different views of information	161
D.2 Observation concerns	162
D.2.1 Domain specialization	162
D.2.2 Comparison with provider-oriented models.....	163
D.2.3 Observation discovery and use	164
D.2.4 Observations, interpretations, simulations	164
D.3 Sample, Sampling concerns	165
D.3.1 Sample as observation-collector	165
D.3.2 Observation feature(s)-of-interest.....	165
D.3.3 Processing chains and intermediate features-of-interest.....	166
D.4 Observations and Coverages.....	166
Annex E (informative) Detailed package overview diagrams	169
E.1 General	169
E.2 Abstract Observation Core — Overview.....	169
E.3 Basic Observations — Overview	169
E.4 Abstract Sample Core — Overview	170
E.5 Basic Samples — Overview	171
Bibliography.....	172

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).

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).

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.

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 287, *Geographic Information*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement), and in collaboration with the Open Geospatial Consortium (OGC).

This second edition cancels and replaces the first edition (ISO 19156:2011), which has been technically revised.

The main changes are as follows:

- the UML model and 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 ISO 19156:2011, but certain carefully designed improvements and clarifications for the intended use have been included;
- the Sample model has been refined: given the integral nature of the Sample model, it has been decided to include that term in the name of the document;
- Annex C has been added listing the changes between ISO 19156:2011 and this document.

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.

Introduction

This document 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 W3C Spatial Data on the Web Best Practices.^[31] This includes several new standards for describing and publishing sensors and observations, such as the OGC SensorThings API^[22] and the W3C/OGC Semantic Sensor Network Ontology.^[28] This second edition of ISO 19156 (now named "Observations, Measurements and Samples", or abbreviated to "OMS") is informed by these recent developments. The focus of revising ISO 19156:2011 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 in this document is derived from the previous edition published by Open Geospatial Consortium as OGC 10-004r3, and also ISO 19156:2011. A technical note describing the changes in comparison to ISO 19156:2011 is provided in Annex C.

The name and contact information of the maintenance agency for this document can be found at [www.iso.org/maintenance agencies](http://www.iso.org/maintenance_agencies).

Geographic information — Observations, measurements and samples

1 Scope

This document 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 document 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 document 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.

2 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, *Geographic information — Conceptual schema language*

ISO 19107, *Geographic information — Spatial schema*

ISO 19108, *Geographic information — Temporal schema*

3 Terms and definitions

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

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

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://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 19123-1:—,¹ 3.1.8]

3.3

data type

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

EXAMPLE Integer, Real, Boolean, String and Date.

Note 1 to entry: 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: All elements within a domain (set) are of a given type.

[SOURCE: ISO 19109:2015, 4.8, modified — Original Note 1 to entry has been replaced with a new Note 1 to entry.]

3.5

domain feature

feature of a type defined within a particular application domain

Note 1 to entry: This can be contrasted with observations and sampling features, which are features of types defined for cross-domain purposes.

3.6

ex situ

off-site

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

Note 1 to entry: Opposite of in situ (on-site).

Note 2 to entry: An example of ex situ and direct is measuring a patient's temperature with a mercury thermometer in a blood sample.

Note 3 to entry: An example of ex situ and remote is measuring a patient's temperature with an infra-red thermometer pointed at the blood sample.

3.7

feature

abstraction of real-world phenomena

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

¹ Under preparation. Stage at the time of publication: ISO/FDIS 19123-1:2023.

[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

3.10

in situ

on-site

referring to the study, maintenance or conservation of a specimen or population without removing it from its natural surroundings

Note 1 to entry: Opposite of ex situ (off-site).

Note 2 to entry: An example of in situ and direct is measuring a patient's temperature with a mercury thermometer in the patient's rectum.

Note 3 to entry: An example of in situ and remote is measuring a patient's temperature with an infra-red thermometer at a distance.

3.11

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]

3.12

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, with the value provided as the result

3.14

observation result

estimate of the value of a property determined through a known observation procedure

3.15

observer

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

Note 1 to entry: An observer is an instance of a sensor, instrument, implementation of an algorithm or a being such as a person.

3.16

procedure

specified way to carry out an activity or a process

[SOURCE: ISO 9000:2015, 3.4.5, modified — Note 1 to entry has been deleted.]

3.17

process

set of interrelated or interacting activities that use inputs to deliver an intended result

[SOURCE: ISO 9000:2015, 3.4.1, modified — Notes 1-6 have been deleted.]

3.18

property

facet or attribute of an object referenced by a name

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

Note 1 to entry: In some communities, the observed property is referred to as the measurand.

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

3.19

property type

characteristic of a feature type

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

Note 1 to entry: The value for an instance of an observable property type can be estimated through an act of observation.

Note 2 to entry: In chemistry-related applications, the term "determinand" or "analyte" is often used.

3.20

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.

3.21

range

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

Note 1 to entry: This is consistent with the more generic definition of "range" in ISO 19107.

[SOURCE: ISO 19123-1:—, 3.1.47]

3.22

sample

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

3.23

sampler

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

3.24

sensor

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

[SOURCE: JCGM 200:2012, 3.8, modified — EXAMPLES and NOTE deleted.]

3.25

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.

3.26

unit of measure

reference quantity chosen from a unit equivalence group

Note 1 to entry: In positioning services, the usual units of measurement are either angular units or linear units. Implementations of positioning services shall clearly distinguish between SI units and non-SI units. When non-SI units are employed, it is required that their relation to SI units be specified.

[SOURCE: ISO 19116:2019, 3.29]

3.27

value

element of a type domain

Note 1 to entry: A value considers a possible state of an object within a class or type (domain).

Note 2 to entry: A data value is an instance of a datatype, a value without identity.

Note 3 to entry: 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.

[SOURCE: ISO/IEC 19501:2005, 0000_5, modified — Notes 1- 3 to entry have been added.]

4 Document conventions

4.1 Abbreviated terms and acronyms

CIS	Coverage Implementation Schema
EO	Earth observation
GFM	General Feature Model
GML	Geography Markup Language
INSPIRE	Infrastructure for Spatial Information in Europe
O&M	Observations and measurements

OMS	Observations, measurements and samples (this document)
OGC	Open Geospatial Consortium
RDA	Research Data Alliance
SensorML	OGC Sensor Model Language
SOS	OGC Sensor Observation Service
STA	OGC SensorThings API
SWE	OGC Sensor Web Enablement
UML	Unified Modeling Language
UoM	unit of measure
URI	Uniform Resource Identifier
XML	Extensible Markup Language
2-D	two-dimensional
3-D	three-dimensional

4.2 Schema language

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

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. The stereotype «FeatureType» states that a class is an instance of the «metaclass» 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.

4.3 Model element names

This document 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 be applied 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 document to common terminology in related application domains is provided in Annex B.

4.4 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 document to be easily located by implementers.

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

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

- a) **/req/obs-cpt**: Conceptual Observation schema;
- b) **/req/obs-core**: Abstract Observation Core;
- c) **/req/obs-basic**: Basic Observations;
- d) **/req/sam-cpt**: Conceptual Sample schema;
- e) **/req/sam-core**: Abstract Sample core;
- f) **/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 attributes 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 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, when the cardinality is constrained within one model package.
- **{Attribute/Association Name}-con**: Additional constraints. As these sometimes pertain to multiple attributes or associations, this part of the name can become more complex.

Individual requirements are case-sensitive, following UML naming conventions. Requirements pertaining to classes contain the class name in UpperCamelCase. Requirements pertaining to attributes or associations utilize the attribute name or 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 document to be easily located by implementers.

4.5 Requirements classes

Each statement (requirement or recommendation) in this document 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 shall be satisfied. Hence, the requirements class is the unit of re-use and dependency.

Dependency to another requirements 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.

4.6 Conformance classes

Conformance to this document is possible at a number of levels, specified by conformance classes in accordance with 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 shall be passed. Each conformance class tests conformance to a set of requirements packaged in a requirements class.

4.7 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 Uniform Resource Identifier (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>

4.8 Associations in UML context diagrams

The UML model described in this document is rather complex. To keep the text size readable in the UML, context diagrams of this document only display certain associations of each class. Please refer to the context diagram of a particular class to see all associations of that class. All associations of the classes in each package are also shown in the detailed package overview diagrams in Annex E.

5 Conformance

5.1 Overview

Clauses 8 to 13 of this document use 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 document.

The UML model in this document 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.

5.2 Conformance classes

The conformance rules for Models in general are described in ISO 19109. Application Schemas also claiming conformance to this document shall also conform to the rules specified in Clauses 8 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 (OMS) conformance. Table 1, Table 2, Table 3, Table 4, Table 5 and Table 6 list all of these classes by package, including their relative identifiers and the corresponding subclauses of the Abstract Test

Suite. The full URIs of the conformance classes are formed by prefixing the relative URI path as described in 4.7.

Table 1 — Conceptual Observation schema conformance classes

Conformance class	Identifier	Subclause
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	Subclause
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
Abstract Observation Core – AbstractObservationCollection	/conf/obs-core/AbstractObservationCollection	A.2.10

Table 3 — Basic Observations conformance classes

Conformance class	Identifier	Subclause
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	Subclause
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	Subclause
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	Subclause
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

6 Packaging, requirements and dependencies

6.1 Requirements

As OMS implementations often seamlessly integrate with existing data ecosystems, a very flexible requirements and conformance structure is defined. This structure enables 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 Some providers only serve information on Observable Properties or Monitoring Facilities, while relying on other partners to provide information on measurement procedures. These can 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^[25] 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.

- a) For each UML Package, both a Requirements Class as well as a conformance class have been defined.
- b) Additional Requirements Classes have been created for each Class appearing in the data model. Conformance Classes are added accordingly to enable grouping of the former and support references to either a group or an individual Requirement Class depending on the need.
- c) Thematic domains can 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 Observations and Measurements (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.

6.2 UML

6.2.1 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.

- a) **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, which, although they have not been created under the OMS model, do provide concepts sharing the same semantic meaning as the concepts from the Conceptual models.
- b) **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.
- c) **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.

6.2.2 UML package dependencies

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

Table 7 — UML package level dependencies

OMS Package	Package	International Standard	Classes
Conceptual Observation schema	Any type	ISO 19103:2015	Any
Conceptual Observation schema	Temporal Objects	ISO 19108:2002	TM_Object
Conceptual Observation schema	Name types	ISO 19103:2015	GenericName
Abstract Observation Core	Conceptual Observation schema	ISO 19156:2023 (this document)	TM_Instant, TM_Period via the Temporal

OMS Package	Package	International Standard	Classes
			Objects dependency
Abstract Observation Core	General Feature Model	ISO 19109:2015	Feature concepts
Abstract Observation Core	Text	ISO 19103:2015	CharacterString
Basic Observations	Abstract Observation Core	ISO 19156:2023 (this document)	
Basic Observations	Web environment	ISO 19103:2015	URI
Basic Observations	Geometry	ISO 19107:2019	Geometry
Conceptual Sample schema	Any type	ISO 19103:2015	Any
Conceptual Sample schema	Temporal Objects	ISO 19108:2002	TM_Object
Conceptual Sample schema	Name types	ISO 19103:2015	GenericName
Conceptual Sample schema	Conceptual Observation schema	ISO 19156:2023 (this document)	Observation, Procedure
Abstract Sample Core	Conceptual Sample schema	ISO 19156:2023 (this document)	
Abstract Sample Core	General Feature Model	ISO 19109:2015	Feature concepts
Abstract Sample Core	Geometry	ISO 19107:2019	Geometry
Abstract Sample Core	Text	ISO 19103:2015	CharacterString
Abstract Sample Core	Abstract Observation Core	ISO 19156:2023 (this document)	NamedValue
Basic Samples	Abstract Sample core	ISO 19156:2023 (this document)	
Basic Samples	Web environment	ISO 19103:2015	URI
Basic Samples	Measure types	ISO 19103:2015	Measure

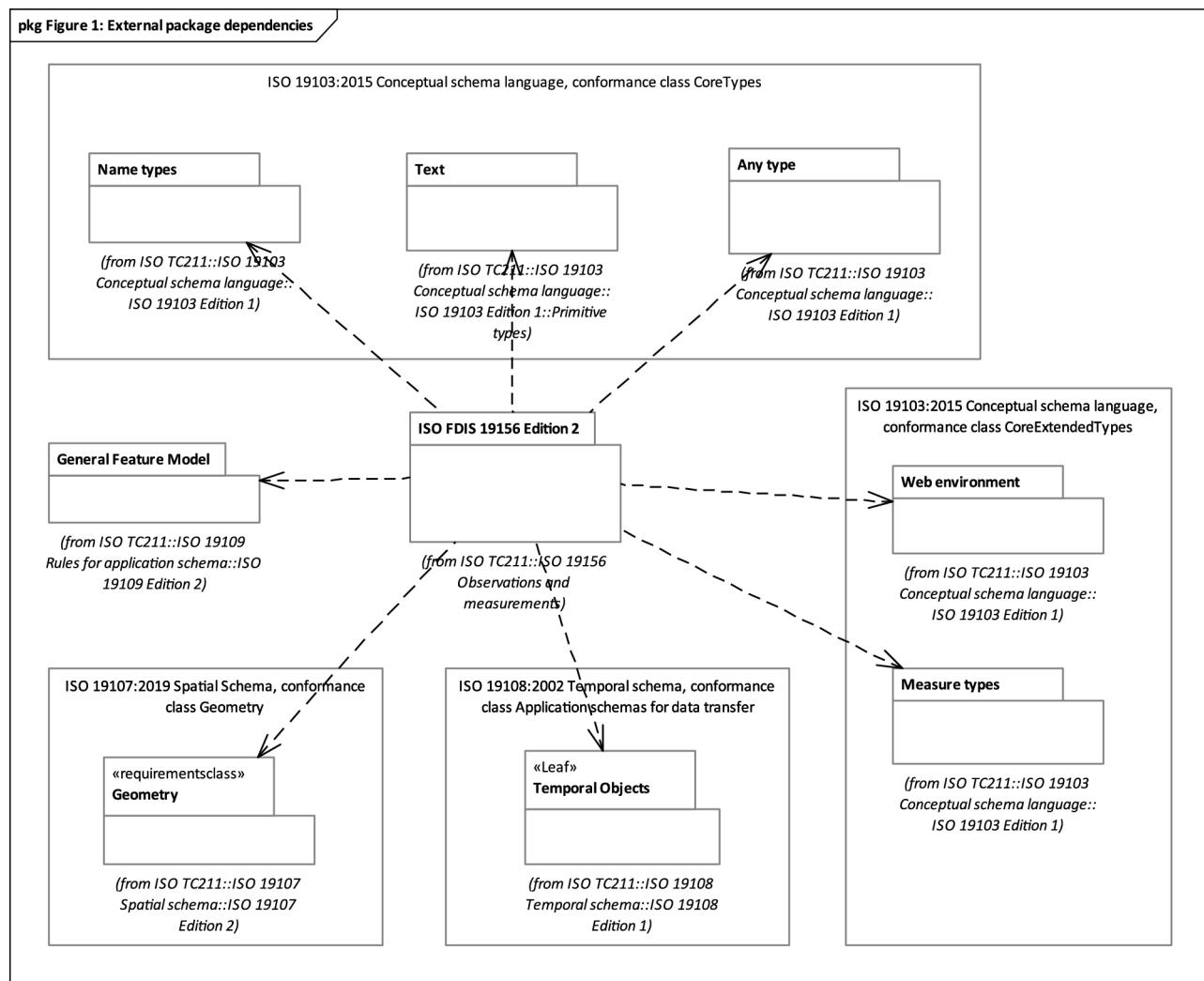


Figure 1 — External UML package dependencies

6.3 Note on the use of "Any"

The UML models defined in this document make extensive use of the Any interface defined in the 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 a digital representation of an appropriate concept. If 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

EXAMPLE 1 Reference to SWEET Ontology:

<http://sweetontology.net/realmAtmoBoundaryLayer#planetaryboundarylayer>

EXAMPLE 2 Reference to SensorThings deployment: [https://lubw-frost.docker01.ilt-dmz.iosb.fraunhofer.de/v1.1/Locations\(269\)](https://lubw-frost.docker01.ilt-dmz.iosb.fraunhofer.de/v1.1/Locations(269))

EXAMPLE 3 Reference to the ISO 19115 series Metadata: https://inspire-geoportal.ec.europa.eu/resources/INSPIRE-61494ff5-6fad-11e8-b649-52540023a883_20210415-080302/services/1/PullResults/701-750/43.iso19139.xml

EXAMPLE 4 Reference to an instance of borehole: <https://data.geoscience.fr/id/borehole/BSS001REWW>

EXAMPLE 5 Reference to a hydrostation: <https://iddata.eaufrance.fr/id/HydroStation/Y251002001>

EXAMPLE 6 Reference to a river segment:
<https://iddata.eaufrance.fr/id/WatercourseLinkSequence/A0080300>

EXAMPLE 7 An (embedded) Boolean value as Result.

EXAMPLE 8 An (embedded) SWE DataRecord.

EXAMPLE 9 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](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)

EXAMPLE 10 OMS MaterialSample -> Reference to a rock sample: https://www.geodata.rocks/Samples/SD-5054_1_A_564_7WR_20-40

7 Fundamental characteristics of observations and samples (informative)

7.1 Observation schema

7.1.1 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.

7.1.2 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 a given 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. References [11], [14], [15], [16], [18], [23]) the term “measurement” is used. However, a distinction between measurement and category-observation has been adopted in more recent work^{[12],[17],[24]} 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. the ISO 19115 series or another community agreed one).

7.1.3 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^{[11],[12]} but may be a process chain, human observer, an algorithm, a computation or a 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. Figure 2 (based on representation work done under INSPIRE^[29]) provides a visual overview of this.

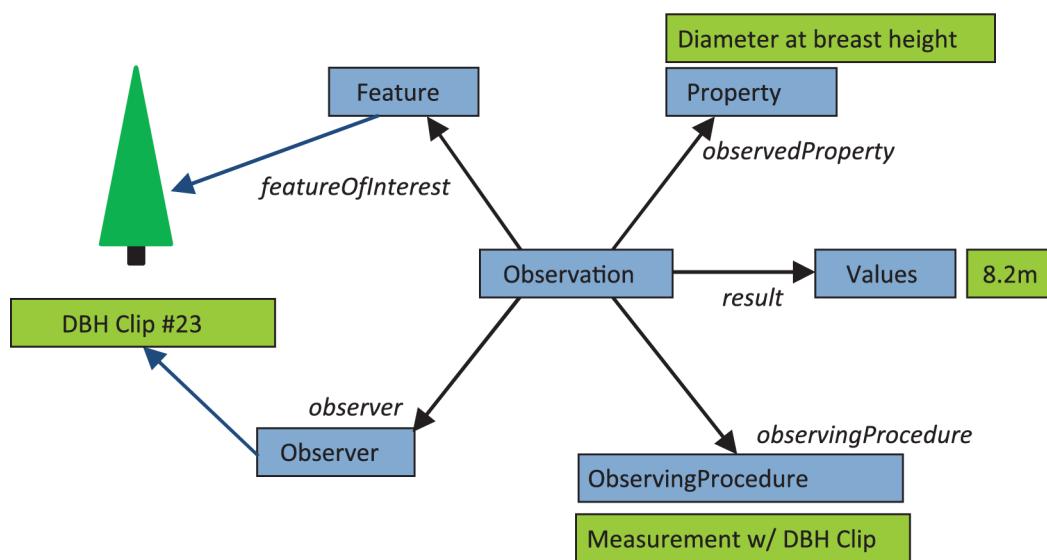


Figure 2 — Properties of an Observation

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 document. This is further elaborated in Clause D.3.

7.1.4 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 can potentially 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 can 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 (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.

7.1.5 Result types

Observation results may have many datatypes, including primitive types like category or measure, but may also 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.

7.1.6 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).

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

7.2 Sample schema

7.2.1 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 the sample would point upwards in the chain of sampled features leading to the 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.

7.2.2 Proximate vs. ultimate feature-of-interest

7.2.2.1 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 is

not always 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:

- a) the observation does not obtain values for the whole of a domain feature;
- b) the observation is performed on a proxy that is not part of the domain feature;
- c) 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.

7.2.2.2 Proximate feature-of-interest embodies a sample design

In some cases, for various reasons, the domain feature is not 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 1, 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 can 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).

7.2.2.3 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 can potentially 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 can be quantified as a voltage, whereas the observed property represented by this voltage is the physiochemical value being observed by the sensor (e.g. pH).

7.2.2.4 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 can 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.

7.2.3 Role of samples

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, etc.) has been established, or 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 are potentially described by the 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 include biodiversity studies, crop seed preservation, etc. In those cases, the material samples considered are called specimen. It is for this reason that the class named "SF_Specimen" in the previous edition of this document has been renamed "MaterialSample" in this edition.

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

NOTE A transient spatial sample, such as a ship-track or flight-line, can 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 is not known when observations are made using the sample.

7.2.4 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 needs to 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 can also be required for the case where one sampling process classifies samples in accordance with multiple criteria.

EXAMPLE 1 When performing observations on populations, these can potentially 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 can involve very different samples, whereby some of these samples can serve purely to provide contextual information pertaining to the sampling event.

EXAMPLE 2 When sampling water from a river, information on the meteorology at the time of sampling can be relevant for the interpretation of measurements obtained on the water sample.

7.2.5 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 can 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.

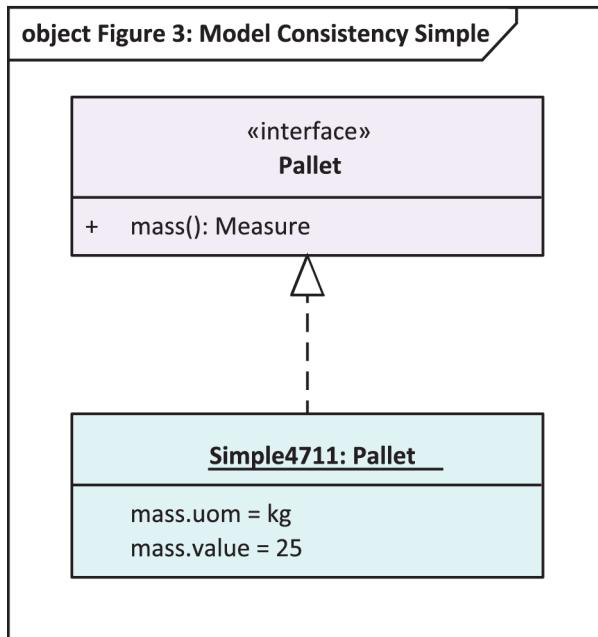
Additional information on provenance, curation and methods of archiving a sample has been delegated to external standards that may be referenced via the ‘metadata’ association that can be provided for all types contained within the Sampling model.

7.3 Alignment between Observation, Sample and domain models

7.3.1 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 3).

EXAMPLE A pallet with the characteristic mass is to be described via a feature model. In the simplest form, an interface “Pallet” can 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.

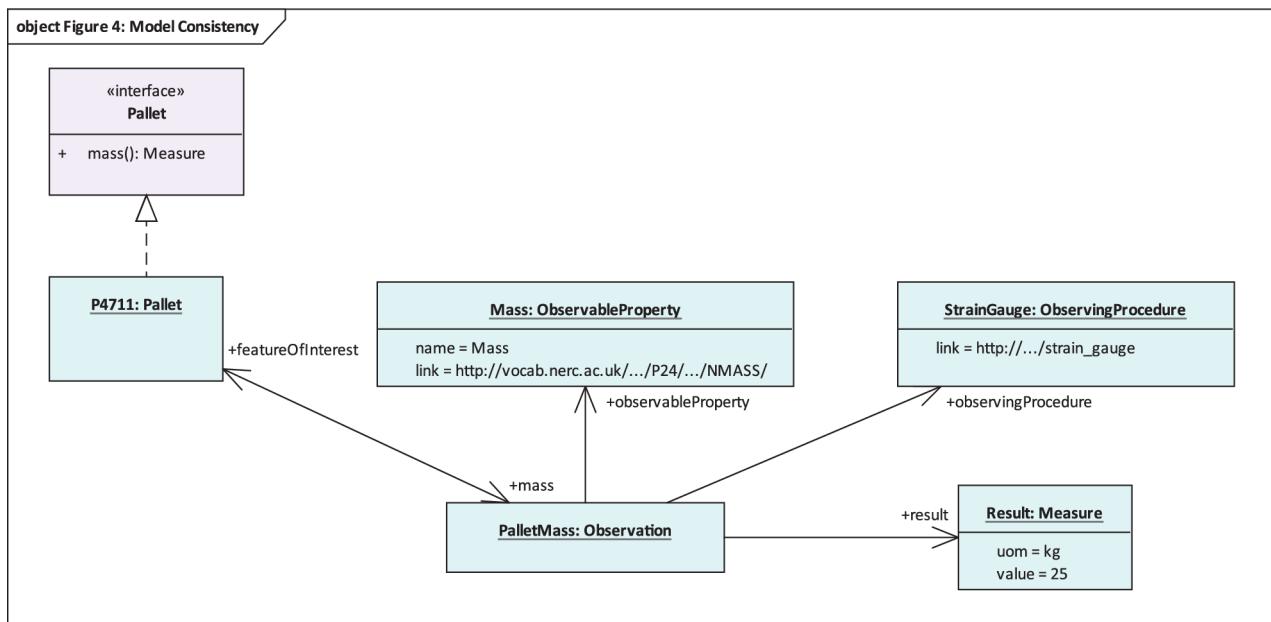


NOTE Simple example for model consistency.

Figure 3 — (Example) Pallet interface

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.

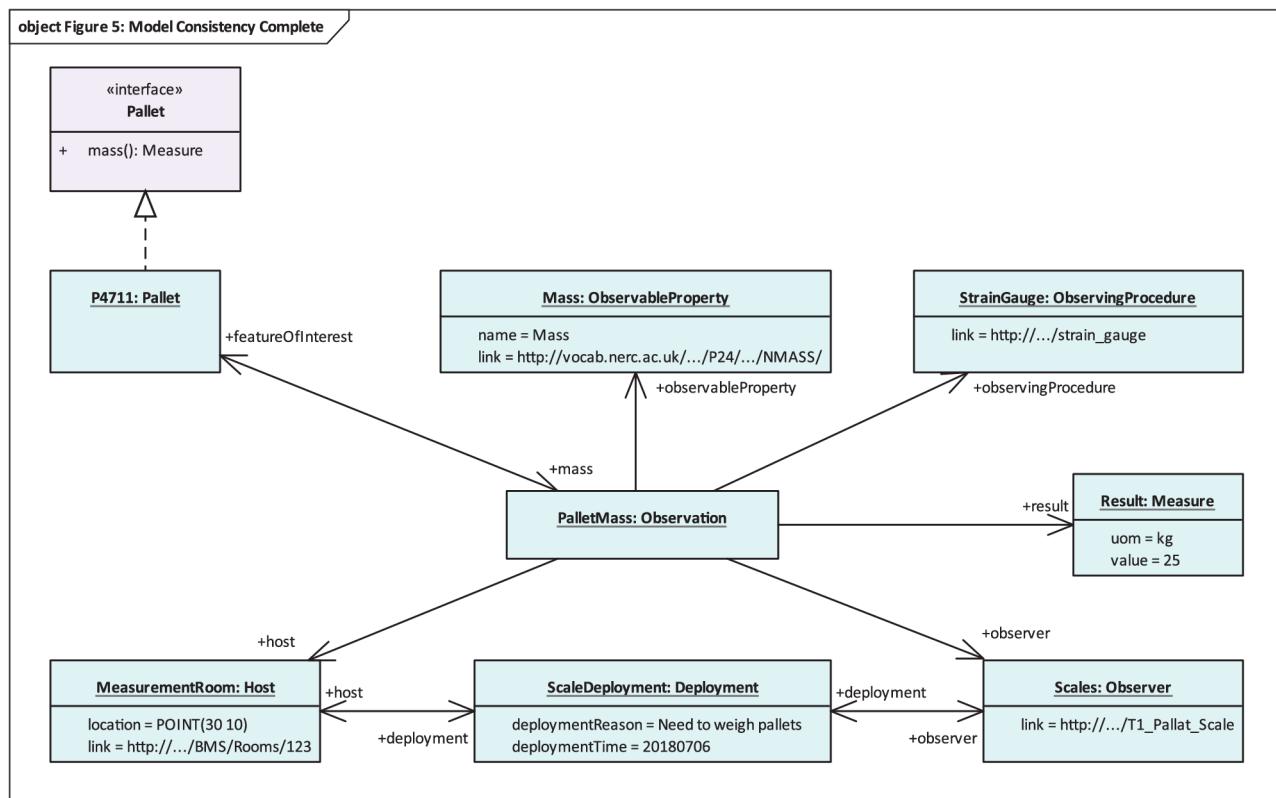
For this purpose, the observation shall have observedProperty “mass”, the result shall be of the type “Measure” and the scale (unit of measure) shall be suitable for mass measurements. Thus, the requirements ensuing from the Pallet Interface are fulfilled, while additional relevant measurement meta-information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



NOTE 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.

Figure 4 — (Example) An observation with consistent properties

Figure 5 shows a complete representation of a mass observation. In addition to the basic information provided with the observation in Figure 4, information on the specific measurement device used is provided together with information on where this device was deployed as the observation was performed.



NOTE For additional context, the Observer, Host and Deployment have been added.

Figure 5 — (Example) An observation with complete properties

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, it is necessary to 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, etc.) is important for water management needs.

7.3.2 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 this Sample 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.

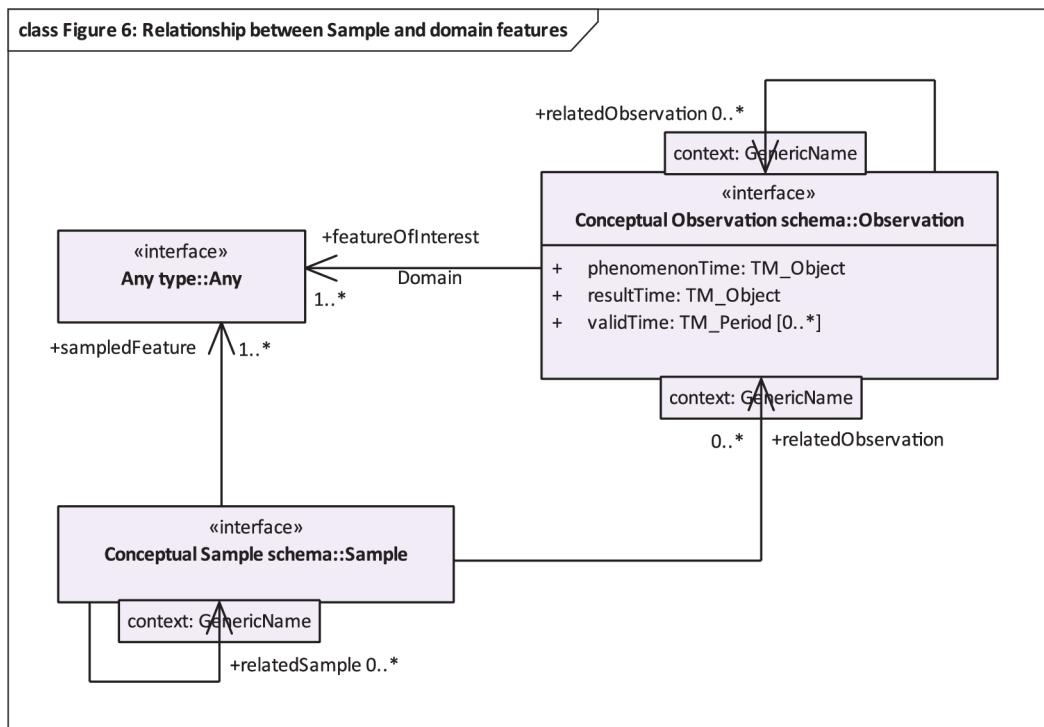


Figure 6 — (Informative) Relationship between Sample and domain features

Both the Sample feature and the domain feature can potentially 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 (in OGC:GWML2 respectively under GW_Well, GW_Aquifer and GW_FluidBody) 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 is the ultimateFeatureOfInterest.

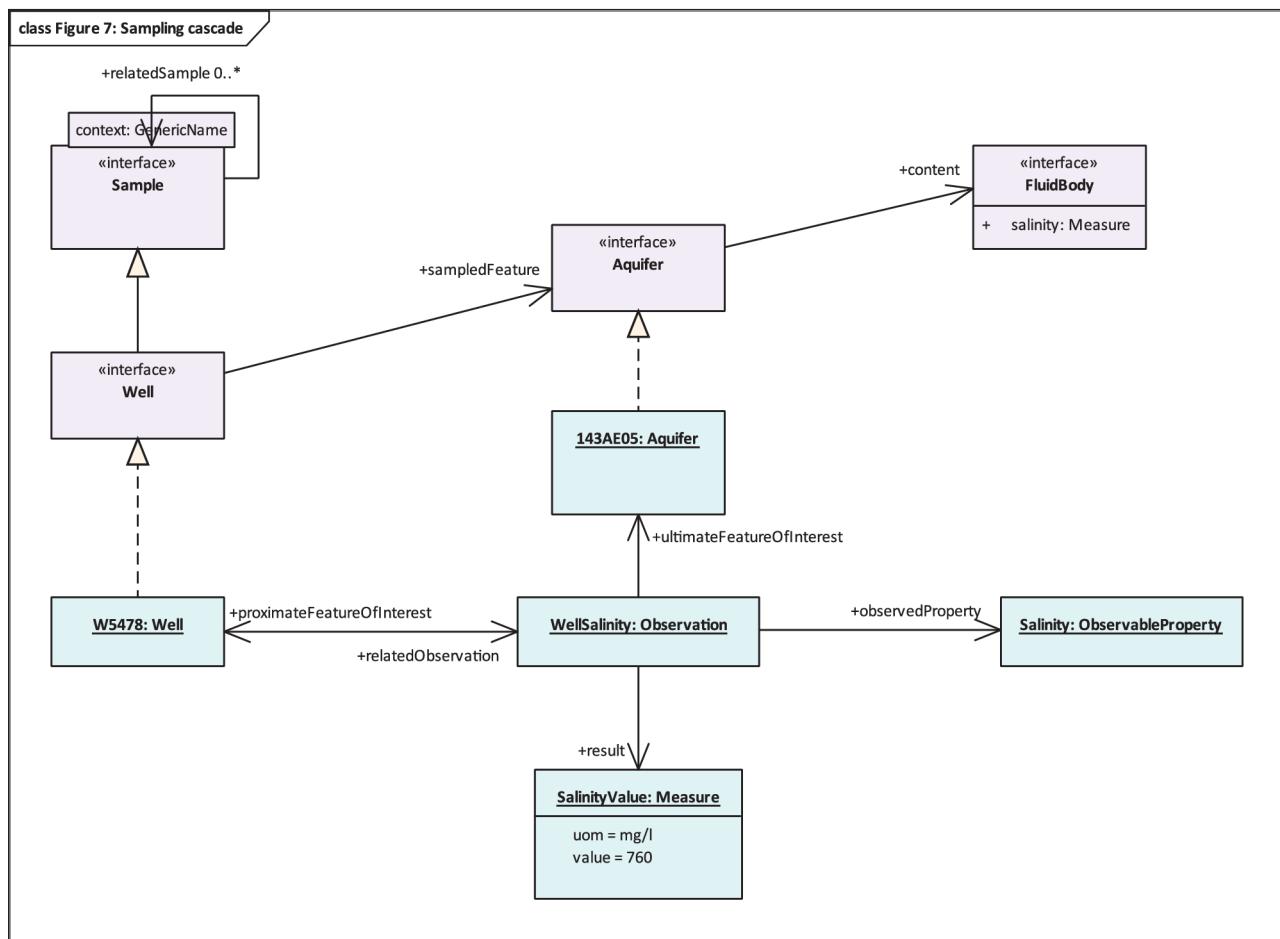


Figure 7 — (Example) Sampling Cascade example including domain features

Depending on the use case, it is advisable to push the modelling choice a step further and instantiate a FluidBody in the system according to the semantics of the domain model (Well, Aquifer, FluidBody). The example is further refined accordingly 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.

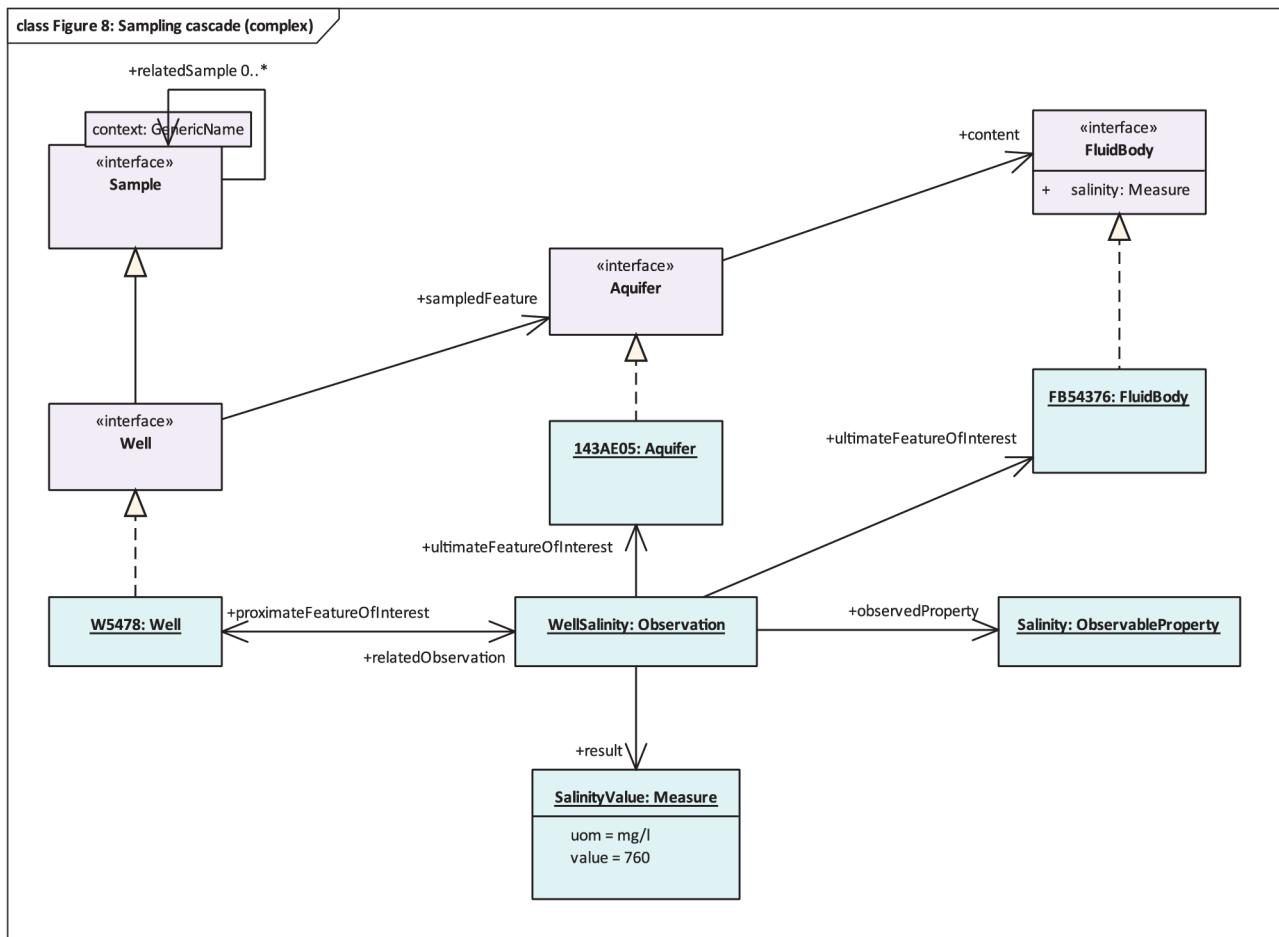


Figure 8 — (Example) Complex Sampling Cascade example referencing external domain feature

8 Conceptual Observation schema

8.1 General

8.1.1 Conceptual Observation model

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

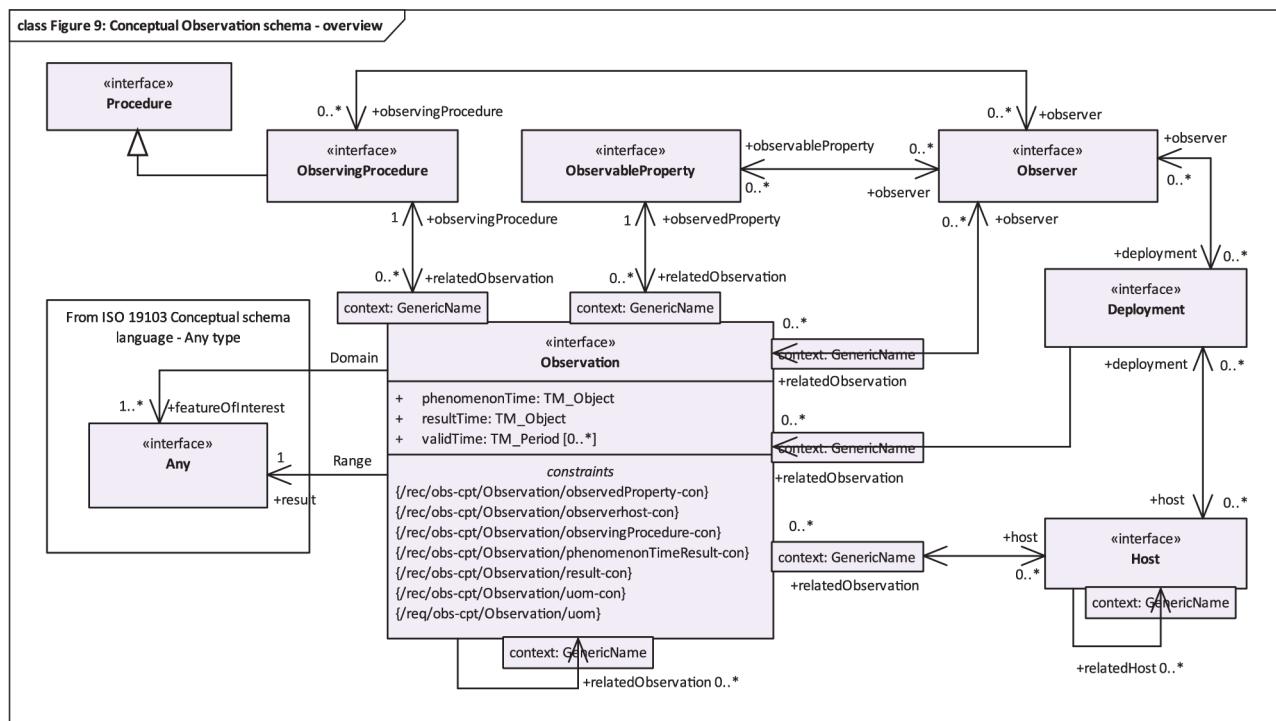


Figure 9 — Conceptual Observation schema overview

8.1.2 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

8.1.3 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.
---	---

8.2 Observation

8.2.1 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

Requirements Class	/req/obs-cpt/Observation
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

8.2.2 Interface Observation

Requirement /req/obs-cpt/Observation/Observation-sem	An Observation shall be defined as 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' and 'simulation' do correspond to this definition. This aspect is further clarified in Clause 7.

8.2.3 Attribute phenomenonTime

Requirement /req/obs-cpt/Observation/phenomenonTime-sem	The time for which 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 of interaction with a real-world feature either by a SamplingProcedure (time at which a Sample has been taken) or by an ObservingProcedure.

NOTE 2 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 .
--	---

8.2.4 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 can be used to estimate the values for phenomena in the future or past. The phenomenonTime is the time to which the result applies, while the resultTime is the time at which the simulation was executed.

Requirement /req/obs-cpt/Observation/resultTime-card	An Observation shall have exactly 1 resultTime .
--	--

8.2.5 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.

8.2.6 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 .
--	--

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 1 The featureOfInterest can be of Any type.

NOTE 2 This object is either the real-world object whose properties are under observation, or it is an object used as a proxy for a real-world object that is not directly observable, as described in 7.2. 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 where objects are created with the intention to sample the real-world object The cardinality of the featureOfInterest association shall be 1 at minimum.
---	---

8.2.7 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 .
--	--

8.2.8 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 can represent the value of any feature property.

NOTE 2 If the observed property is a spatial operation or function, the type of the result can 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 .
--	--

8.2.9 Association observingProcedure

Requirement /req/obs-cpt/Observation/observingProcedure-sem	The ObservingProcedure is 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 .
--	--

8.2.10 Association observer

Requirement /req/obs-cpt/Observation/observer-sem	An Observer 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.
---	--

8.2.11 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.
---	--

8.2.12 Constraint Observer or Host

Recommendation /rec/obs-cpt/Observation/observerhost-con	At least one Observer or Host should be provided.
--	---

8.2.13 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 .
--	--

8.2.14 Constraint suitable ObservableProperty

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

8.2.15 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 .
--	--

8.2.16 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 1 In the case where the result of the Observation is a classification, for which no unit exists, the UoM can be declared as unitless (e.g. referencing the QUDT^[27] <https://qudt.org/vocab/unit/UNITLESS> or UCUM^[19] entry for "no units").

8.3 ObservableProperty

8.3.1 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

8.3.2 Interface ObservableProperty

Requirement /req/obs-cpt/ObservableProperty/ObservableProperty-sem	An ObservableProperty shall be defined as 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, the:

- a) Groundwater Level (1 observable property)

is monitored

- b) with an automated probe (that remains in the ground all year, constituting 1 procedure).

In addition, the groundwater well is revisited in the context of physical campaigns, where the:

- c) Groundwater Level (still the same observable property as above)

is measured, but

- d) with a manual probe (this is a different procedure than used above).

This allows for checking whether the probe needs recalibration.

8.3.3 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.
--	---

8.4 Procedure

8.4.1 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

8.4.2 Interface Procedure

Requirement /req/obs-cpt/Procedure/Procedure-sem	A Procedure shall be defined as a description of steps performed.
--	--

NOTE 1 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.

NOTE 2 The term "process" that was used in ISO 19156:2011 was purposely dropped in this document to avoid unnecessary confusion between the terms "procedure" and "process".

8.5 ObservingProcedure

8.5.1 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

8.5.2 Interface ObservingProcedure

Requirement /req/obs-cpt/ObservingProcedure/ObservingProcedure-sem	An ObservingProcedure shall be defined as the description of steps performed in order to determine the value of an observableProperty by an Observer .
--	---

NOTE 1 Depending on the complexity of the use case, the procedure will be more or less explicitly described. Especially pertaining to historical data, there can be very little or no information available. Information on the recipe that the observer (cook) follows to generate the Observation is ideally included.

NOTE 2 The procedure is often referred to as the method.

NOTE 3 Different observers can follow the same (reusable) procedure for the creation of different observations.

NOTE 4 The procedure is a workflow, protocol, plan, algorithm, or computational method specifying how to make an observation.

NOTE 5 The observing procedure cannot describe a sensor instance, but it can describe the sensor type.

NOTE 6 The term "process" that was used in ISO 19156:2011 has been purposely dropped in this document 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.

8.5.3 Association observer

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

8.6 Observer

8.6.1 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

8.6.2 Interface Observer

Requirement /req/obs-cpt/Observer/Observer-sem	An Observer shall be defined as an identifiable entity that can generate Observations pertaining to an observableProperty by implementing an ObservingProcedure .
--	---

NOTE 1 Different Observers can follow the same (reusable) observing Procedure for the creation of different Observations.

NOTE 2 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.

NOTE 3 An Observer is closely linked with an observableProperty for which it generates results for.

NOTE 4 An Observer can be hosted by one or more Host.

NOTE 5 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, etc. are Observers that are typically mounted on a modern smartphone (which acts as Host). Other examples of sensors include the human eyes.

8.6.3 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.
--	--

8.6.4 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.
--	--

8.6.5 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.
--	---

8.7 Host

8.7.1 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

8.7.2 Interface Host

Requirement /req/obs-cpt/Host/Host-sem	A Host shall be defined as a grouping of Observers for a specific reason.
--	---

NOTE 1 In many use cases, the Host is the environmental monitoring facility.

NOTE 2 The Host can be a platform that hosts a set of sensors.

NOTE 3 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.

8.7.3 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.
--	--

8.7.4 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.
---	--

8.8 Deployment

8.8.1 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

8.8.2 Interface Deployment

Requirement /req/obs-cpt/Deployment/Deployment-sem	A Deployment shall be defined as information on the assignment of an Observer to a Host .
--	--

EXAMPLE 1 Information regarding a sensor being attached to a pole.

EXAMPLE 2 The monitoring facilities pertaining to an environmental monitoring network.

EXAMPLE 3 The description of a ship cruise linking a research vessel with a marine network.

EXAMPLE 4 The participation of a citizen in a citizen-science project involving crowd sensing.

8.8.3 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.
--	--

8.8.4 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
--	---

9 Abstract Observation Core

9.1 General

9.1.1 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
Imports	/req/obs-core/AbstractObservationCollection

9.1.2 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 When providing metadata, using the classes, attributes and associations explicitly modelled in the OMS greatly improves the interoperability compared to using the generic metadata association to include the same information.

9.2 AbstractObservationCharacteristics

9.2.1 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

Requirements class	/req/obs-core/AbstractObservationCharacteristics
Requirement	/req/obs-core/AbstractObservationCharacteristics/pFoI-sem
Requirement	/req/obs-core/AbstractObservationCharacteristics/uFoI-sem
Requirement	/req/obs-basic/ObservationCharacteristics/collection-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

AbstractObservationCharacteristics and AbstractObservation from the Abstract Observation Core are described as a class diagram in Figure 10. The schema is fully described in 9.2 and 9.3.

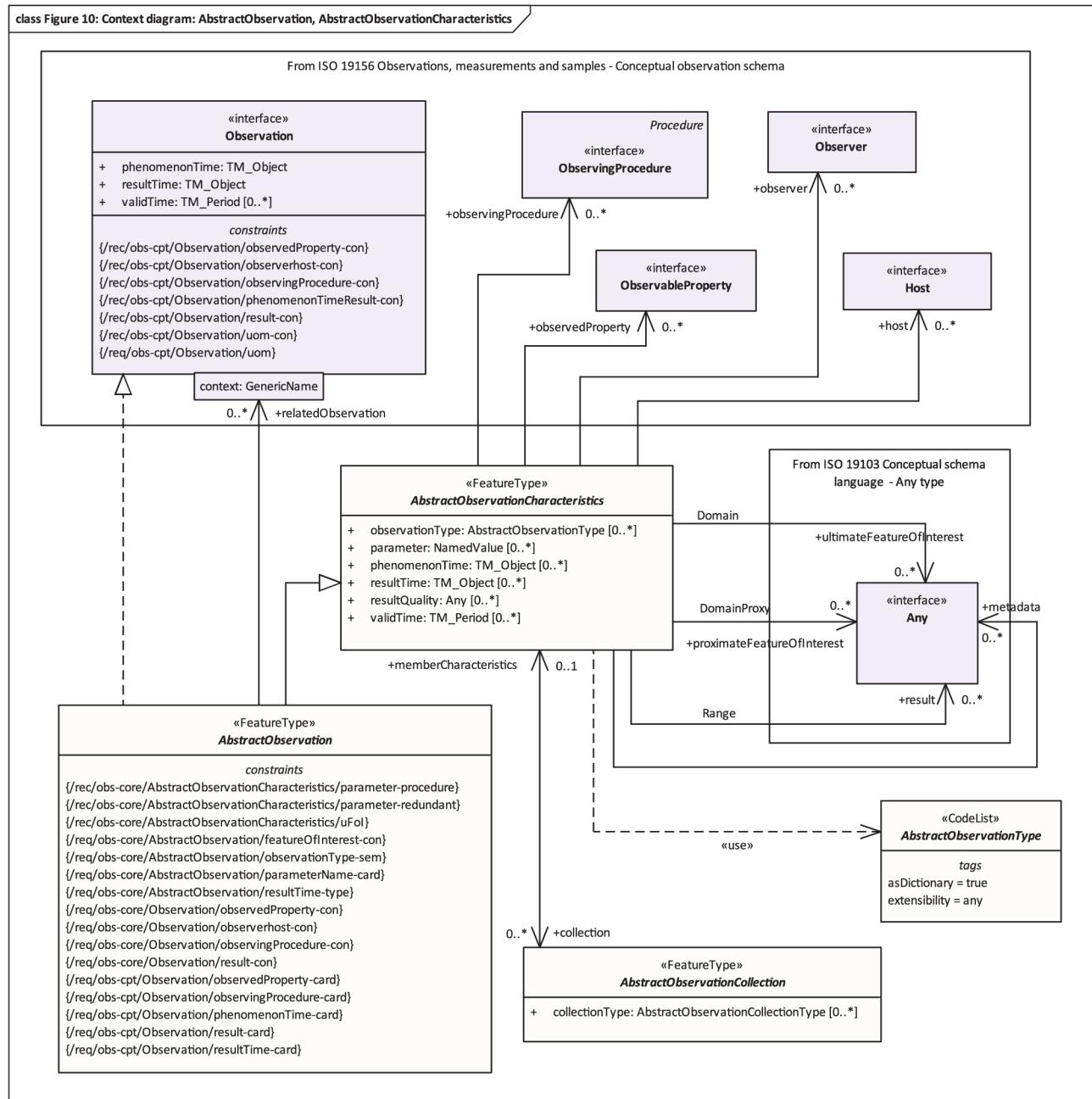


Figure 10 — Context diagram for Abstract Observation Core — AbstractObservationCharacteristics and AbstractObservation

9.2.2 Feature type AbstractObservationCharacteristics

Requirement /req/obs-core/AbstractObservationCharacteristics/AbstractObservationCharacteristics-sem	An AbstractObservationCharacteristics shall be defined as a set of common characteristics used for describing an Observation or a collection of Observations.
---	---

9.2.3 Attribute observationType

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

NOTE 1 Observation type allows describing the formalism, encoding, etc. to be expected when accessing objects associated to the Observation.

NOTE 2 Multiple types can 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).

9.2.4 Attribute parameter

Requirement /req/obs-core/AbstractObservationCharacteristics/parameter-sem	<p>Arbitrary event-specific parameter relevant to the AbstractObservationCharacteristics.</p> <p>If additional parameter information is provided, the property parameter:NamedValue shall be used.</p>
--	--

Recommendation /rec/obs-core/AbstractObservationCharacteristics/parameter-procedure	<p>Parameter should not be used instead of the procedure to describe the steps performed in order to determine the value of the ObservableProperty.</p>
---	--

Recommendation /rec/obs-core/AbstractObservationCharacteristics/parameter-redundant	<p>Parameter should not be utilized to provide information already contained in the model by existing attributes or associations.</p>
---	--

The ObservingProcedure 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 or operator may augment the description of a standard procedure.

EXAMPLE A time sequence of observations of water quality in a well can 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.

NOTE 1 This can 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 2 Parameters that are tightly bound to the procedure can be recorded as part of the procedure description.

9.2.5 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 can be assessed following the procedures in the ISO 19157 series. Multiple measures can be provided.

9.2.6 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 can 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).

9.2.7 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.
---	---

EXAMPLE 1 A river, an aquifer, soil layer, outcrop, a butterfly, a survey area, a room, Abby's car, a specific human being, this document.

EXAMPLE 2 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 3 Pertaining to documents and observations on the consistency of documents, for the Observation "This clause is inconsistent", the **ultimateFeatureOfInterest** is the entire document, while the **proximateFeatureOfInterest** is the specific clause being addressed.

EXAMPLE 4 The determination of the species of the butterfly, in this case the butterfly is the **ultimateFeatureOfInterest**, no **proximateFeatureOfInterest** need be provided.

NOTE 1 The measurement process can 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 <code>/rec/obs-core/AbstractObservationCharacteristics/uFoI</code>	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 2 There will often be a specifiable relationship between the proximate and ultimate feature-of-interest, such as a sampling-chain; see 7.2.2 for examples.

9.2.8 Association collection

Requirement <code>/req/obs-basic/ObservationCharacteristics/collection-sem</code>	An AbstractObservationCollection that is described by these AbstractObservationCharacteristics . If a reference to an AbstractObservationCollection from the AbstractObservationCharacteristics is provided, the association with the role collection shall be used.
---	--

9.3 AbstractObservation

9.3.1 AbstractObservation Requirements Class

Requirements Class	<code>/req/obs-core/AbstractObservation</code>
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	<code>/req/obs-core/AbstractObservationCharacteristics</code>
Requirement	<code>/req/obs-cpt/Observation/phenomenonTime-card</code>
Requirement	<code>/req/obs-cpt/Observation/resultTime-card</code>
Requirement	<code>/req/obs-cpt/Observation/observedProperty-card</code>
Requirement	<code>/req/obs-cpt/Observation/observingProcedure-card</code>
Requirement	<code>/req/obs-cpt/Observation/result-card</code>

Requirements Class	/req/obs-core/AbstractObservation
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

9.3.2 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.
---	---

9.3.3 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 .
---	--

9.3.4 Constraint parameter unique name

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

9.3.5 Constraint proximate or ultimate featureOfInterest.

Requirement /req/obs-core/AbstractObservation/featureOfInterest-con	At least one proximateFeatureOfInterest or ultimateFeatureOfInterest shall be given.
---	--

9.3.6 Constraint Observer or Host

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

9.3.7 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 .
--	---

9.3.8 Constraint suitable ObservableProperty

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

9.3.9 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 .
--	---

9.4 AbstractObservableProperty

9.4.1 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
-------------	--------------------------------

AbstractObservableProperty from the Abstract Observation Core is described as a class diagram in Figure 11. The schema is fully described in 9.4.

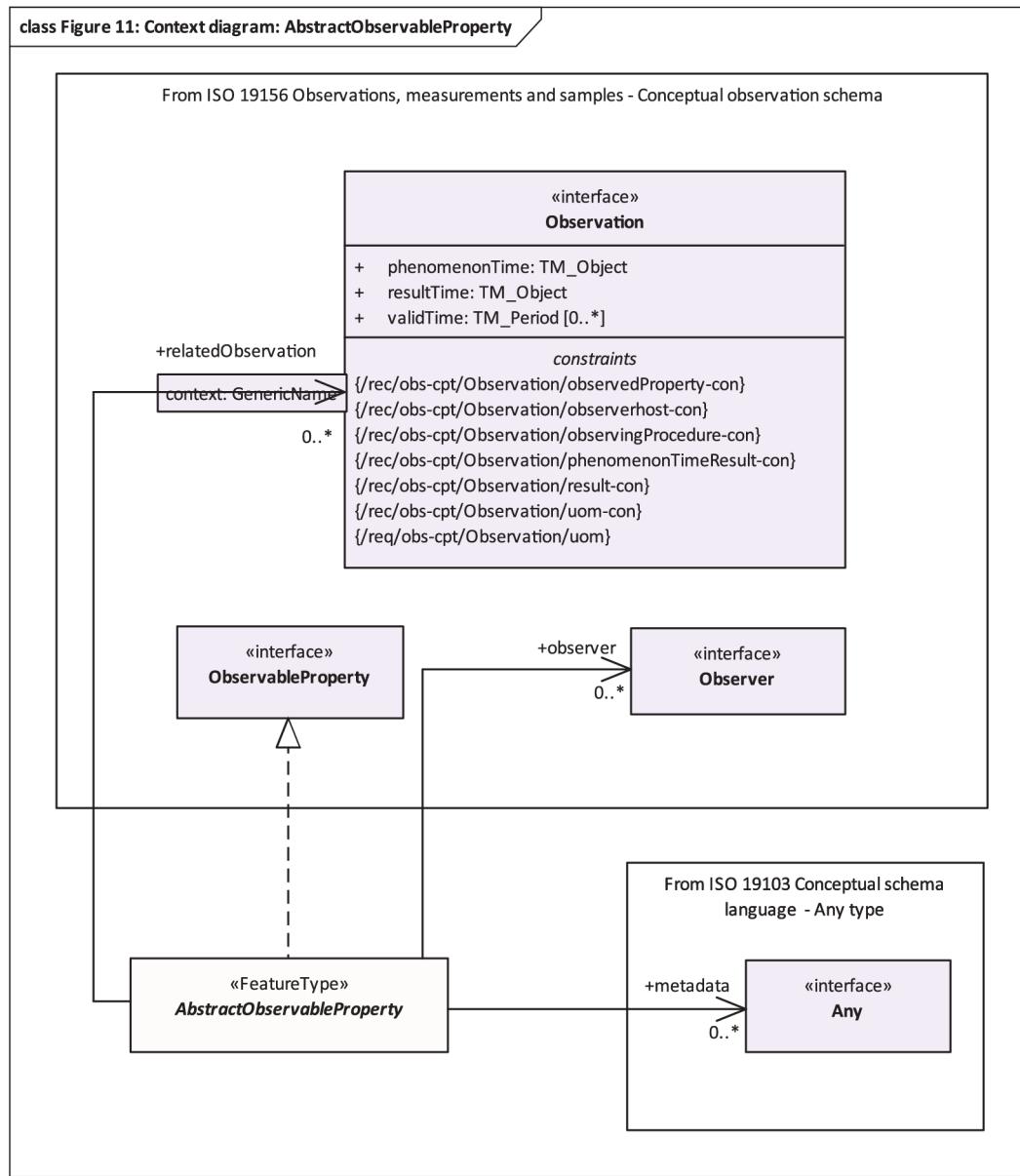


Figure 11 — Context diagram for Abstract Observation Core — `AbstractObservableProperty`

9.5 AbstractObservingProcedure

9.5.1 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

AbstractObservingProcedure from the Abstract Observation Core is described as a class diagram in Figure 12. The schema is fully described in 9.5.

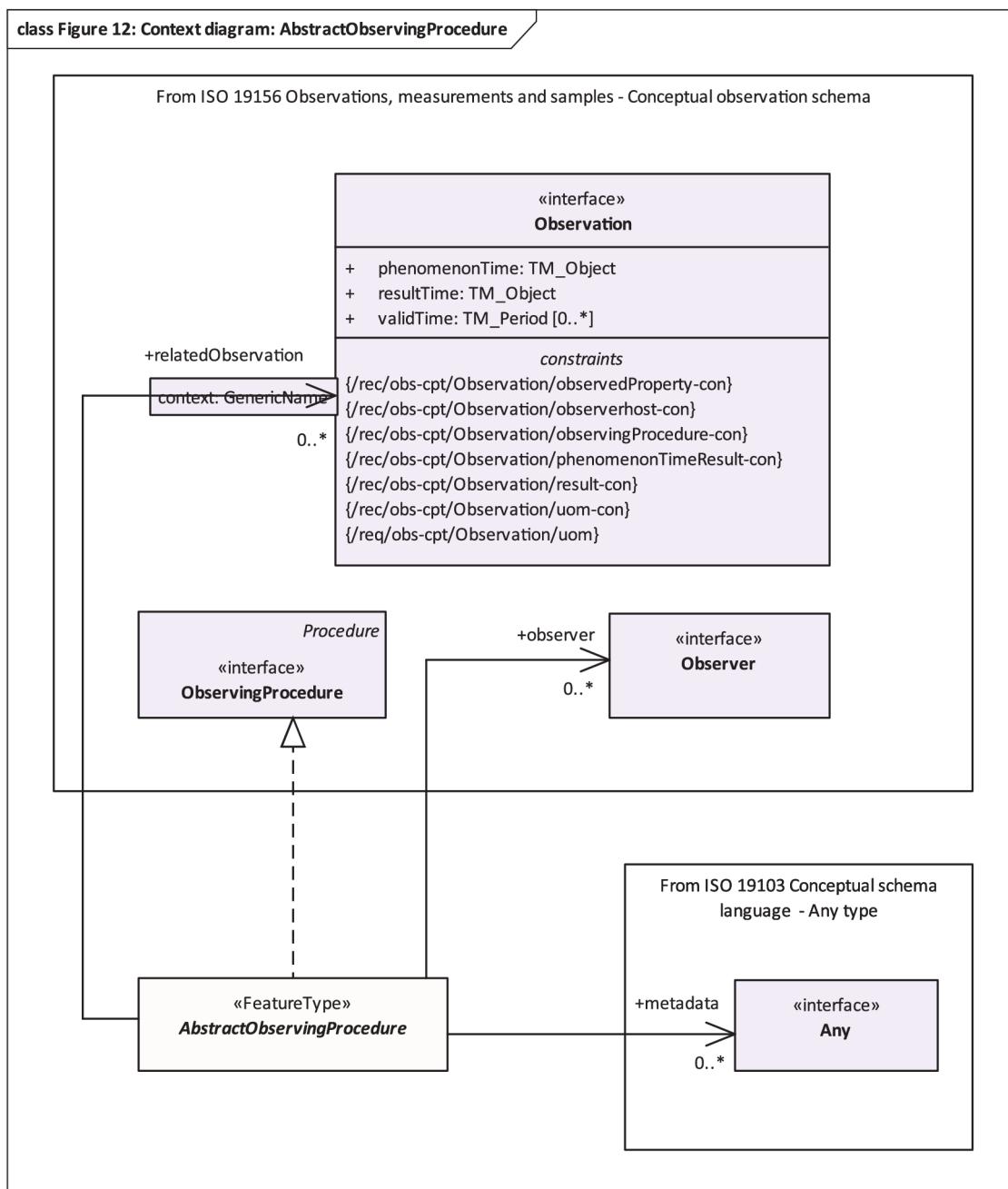


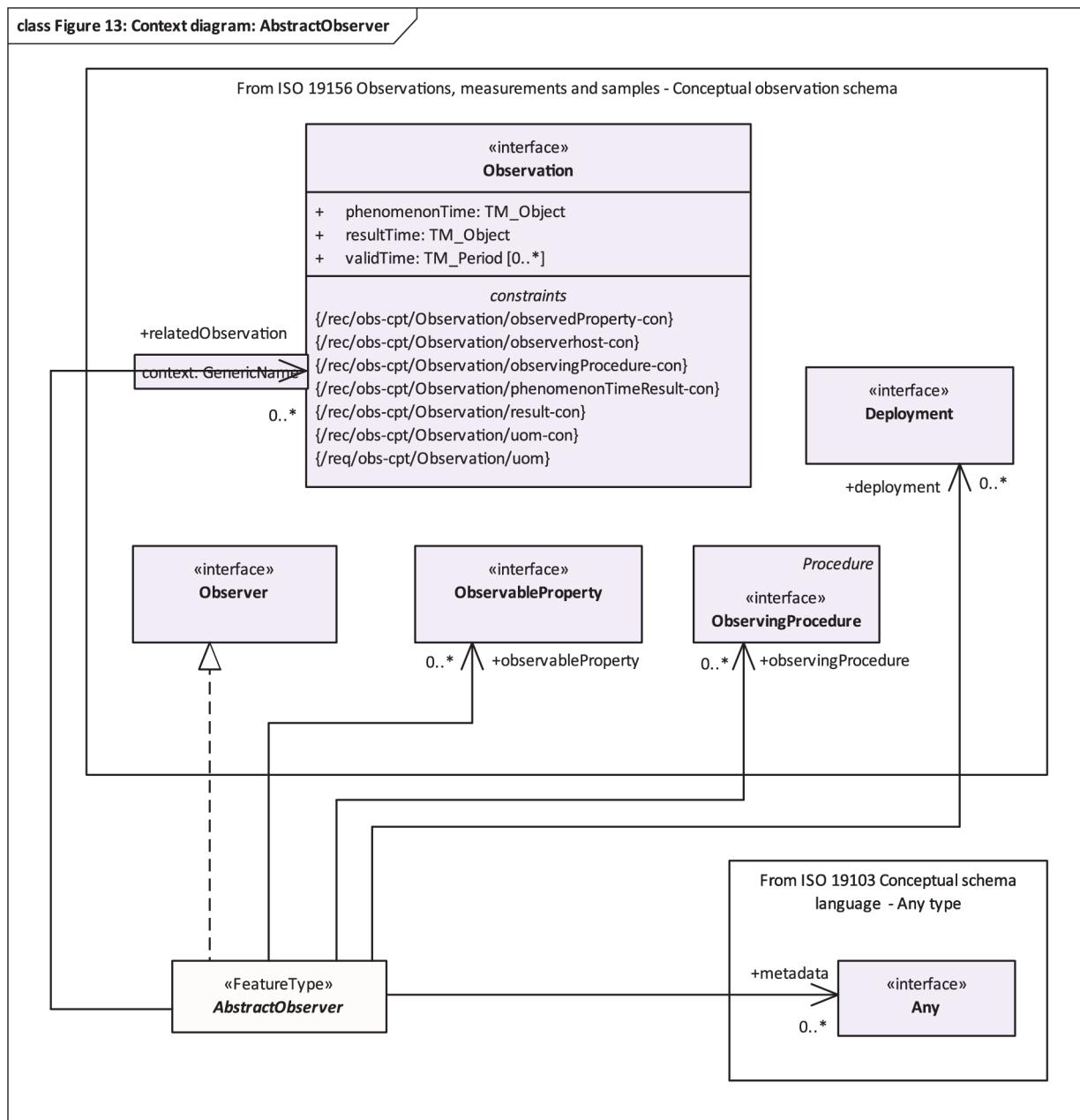
Figure 12 — Context diagram for Abstract Observation Core — AbstractObservingProcedure

9.6 AbstractObserver

9.6.1 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

AbstractObserver from the Abstract Observation Core are described as a class diagram in Figure 13. The schema is fully described in 9.6.

Figure 13 — Context diagram for Abstract Observation Core — `AbstractObserver`

9.7 AbstractHost

9.7.1 AbstractHost Requirements Class

Requirements Class	/req/obs-core/AbstractHost
Target type	Logical model
Name	Abstract Observation Core – <code>AbstractHost</code>

Dependency	ISO 19103:2015, Geographic information — Conceptual schema language, UML2 conformance class
Imports	/req/obs-cpt/Host
Requirement	/req/obs-core/gen/metadata-sem

AbstractHost from the Abstract Observation Core is described as a class diagram in Figure 14. The schema is fully described in 9.7.

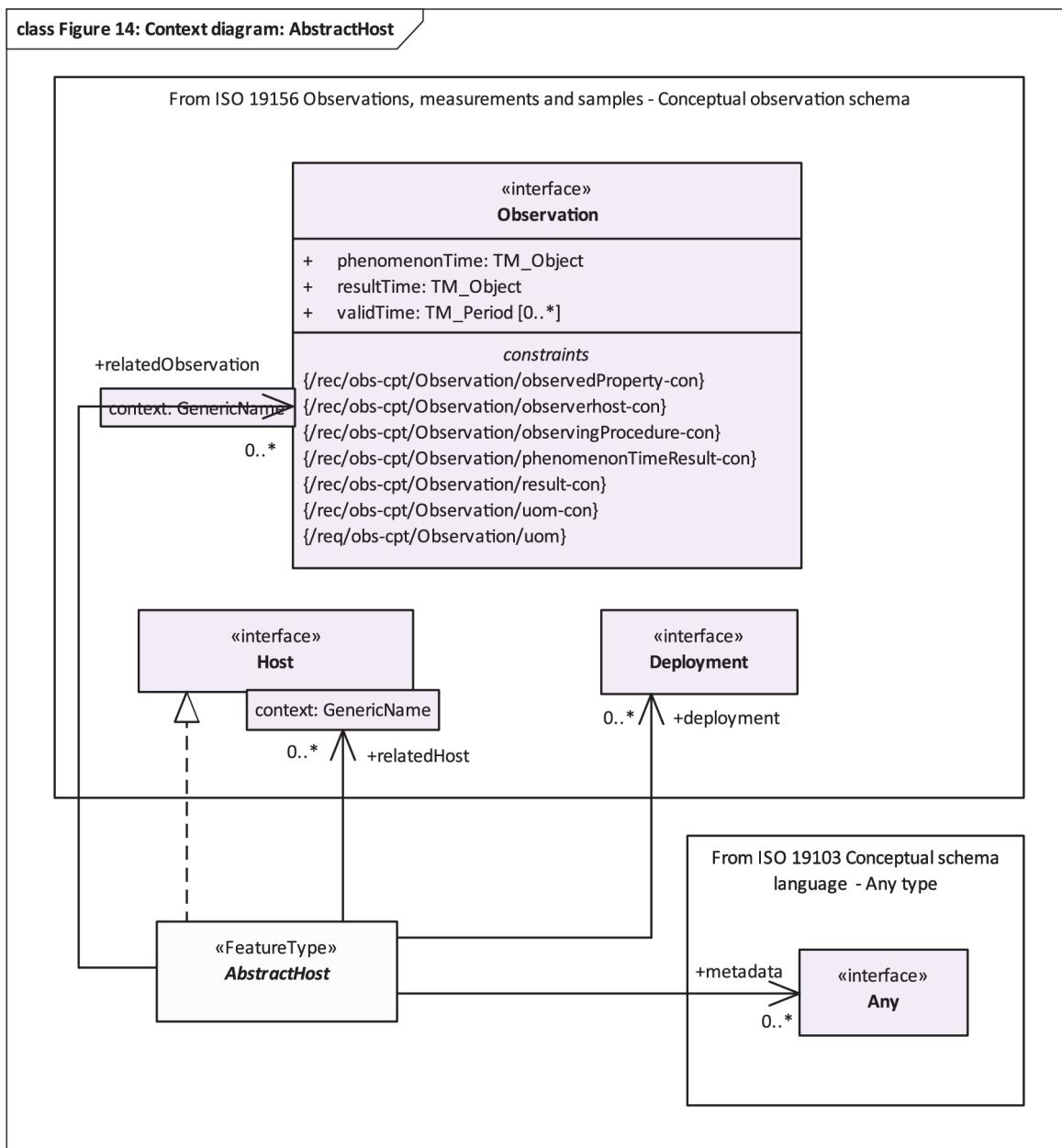


Figure 14 — Context diagram for Abstract Observation Core — AbstractHost

9.8 AbstractDeployment

9.8.1 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

AbstractDeployment from the Abstract Observation Core are described as a class diagram in Figure 15. The schema is fully described in 9.8.

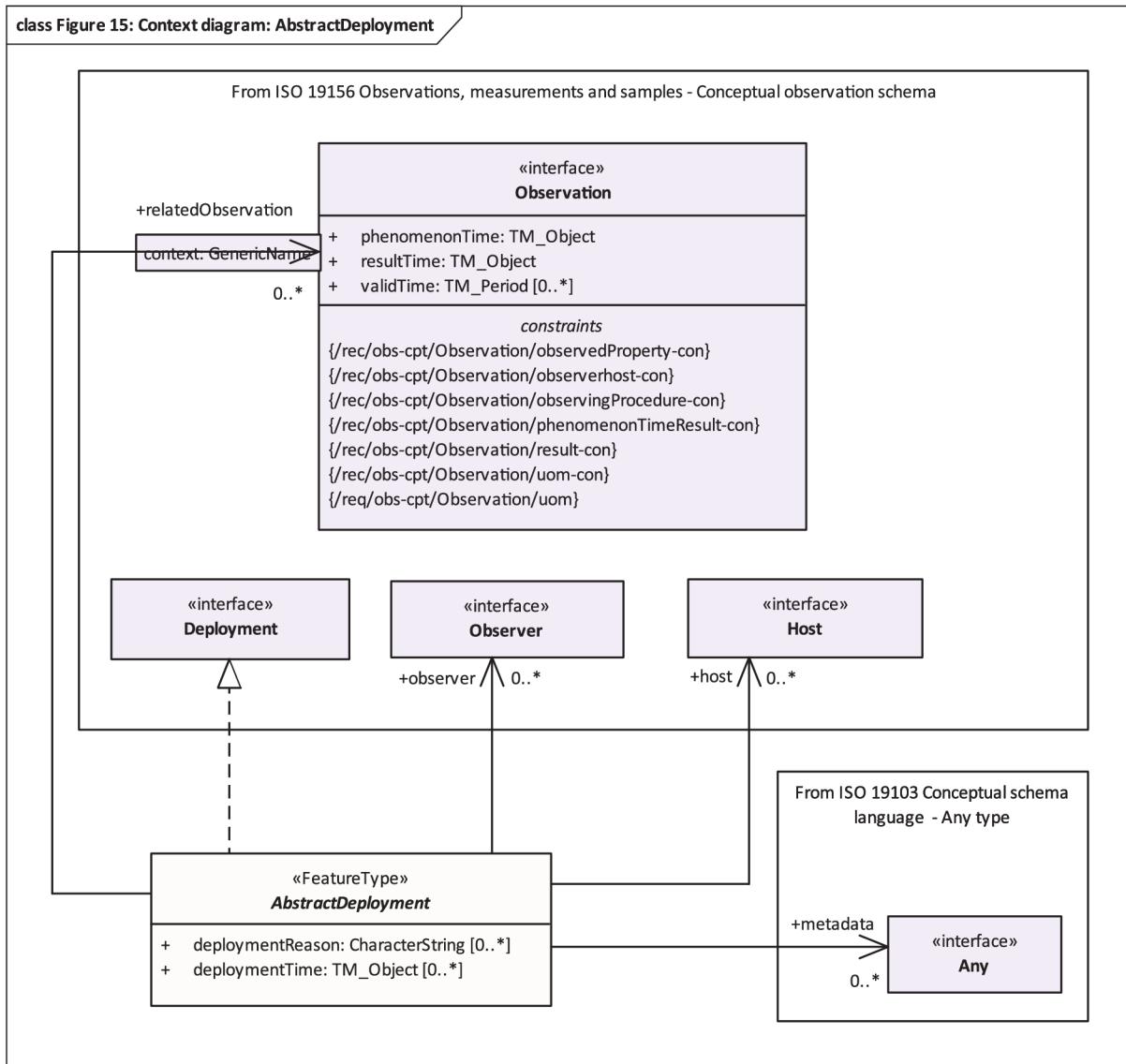


Figure 15—Context diagram for Abstract Observation Core — AbstractDeployment

9.8.2 Attribute deploymentReason

Requirement <code>/req/obs-core/AbstractDeployment/deploymentReason-sem</code>	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.
--	--

EXAMPLE 1 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.

EXAMPLE 2 A sensor is mounted on a building to monitor seismic activities.

EXAMPLE 3 A new sensor type is rolled out within a regional or thematic network due to new legal reporting requirements.

9.8.3 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.
---	---

EXAMPLE 1 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.

EXAMPLE 2 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.

9.9 AbstractObservationCollection

9.9.1 AbstractObservationCollection Requirements Class

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

AbstractObservationCollection from the Abstract Observation Core is described as a class diagram in Figure 16. The schema is fully described in 9.9.

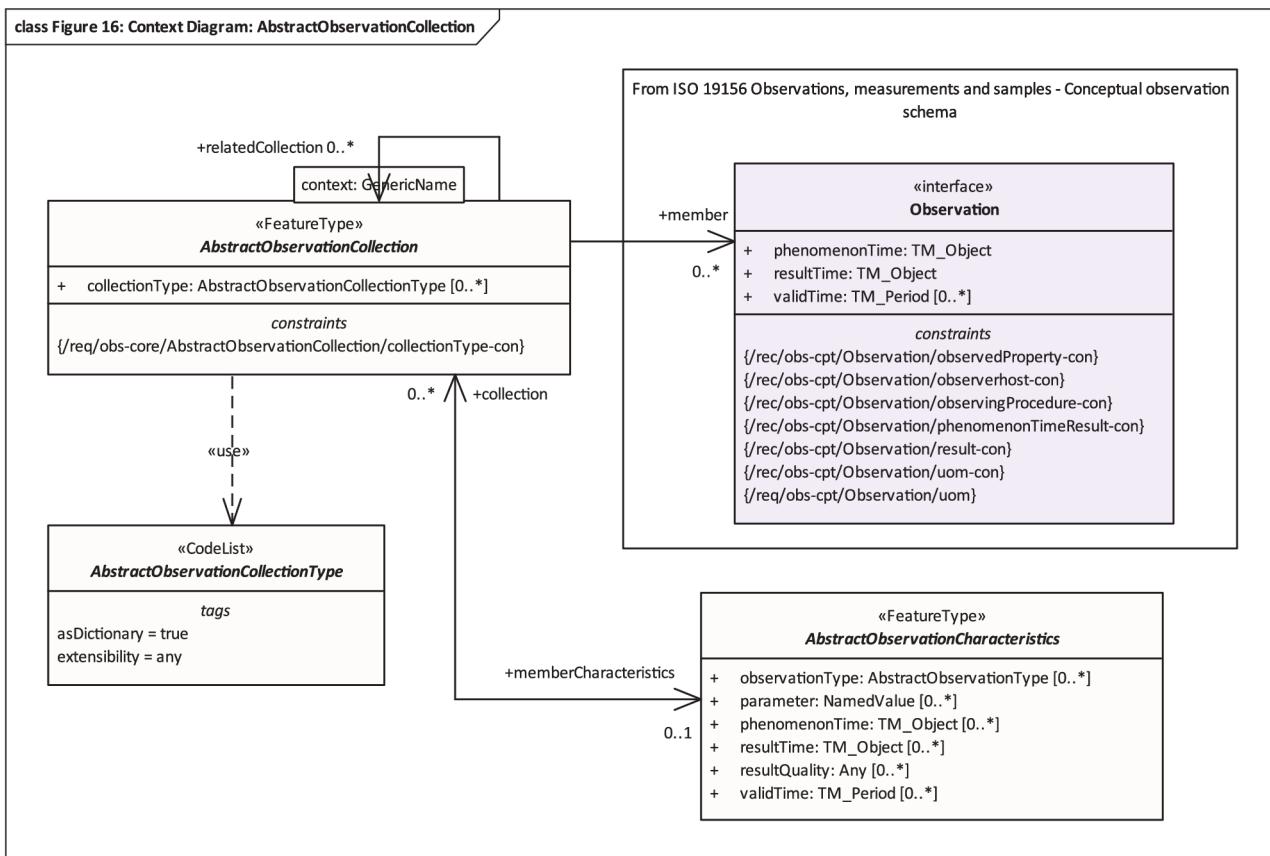


Figure 16— Context diagram for Abstract Observation Core — AbstractObservationCollection

9.9.2 Feature type AbstractObservationCollection

Requirement /req/obs-core/AbstractObservationCollection/AbstractObservationCollection-sem	An AbstractObservationCollection shall be defined as a collection of similar Observations .
---	---

9.9.3 Attribute collectionType

Requirement /req/obs-core/AbstractObservationCollection/collectionType-sem	Information on the type of the AbstractObservationCollection . If information on the collection type is provided, the attribute collectionType:AbstractObservationCollectionType shall be used.
--	---

Requirement /req/obs-core/AbstractObservationCollection/collectionType-con	If the collectionType is provided, property values of the associated Observation and AbstractObservationCharacteristics instances shall comply with the constraints defined for this collectionType value.
--	--

9.9.4 Association member

Requirement /req/obs-core/AbstractObservationCollection/member-sem	An Observation that is part of this AbstractObservationCollection . If a reference to a member Observation is provided, the association with the role member shall be used.
--	--

9.9.5 Association memberCharacteristics

Requirement /req/obs-core/AbstractObservationCollection/memberCharacteristics-sem	Information on AbstractObservationCharacteristics of Observations contained within the AbstractObservationCollection . If a reference to AbstractObservationCharacteristics pertaining to the collection members is provided, the association with the role memberCharacteristics shall be used.
---	--

9.9.6 Association relatedCollection

Requirement /req/obs-core/AbstractObservationCollection/relatedCollection-sem	An AbstractObservationCollection the AbstractObservationCollection is related to. If a reference to a related AbstractObservationCollection 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.
---	--

9.10 NamedValue

9.10.1 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

9.10.2 Data type NamedValue

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

9.10.3 Attribute name

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

NOTE Using well-governed sources for the value of the name enhances reusability.

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

9.10.4 Attribute value

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

NOTE In concrete realizations, the type "Any" can be substituted by a suitable concrete type, such as CI_ResponsibleParty or Measure.

9.11 Codelists

9.11.1 AbstractObservationType

The code list AbstractObservationType can be specialized as required to more precisely define the 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.
---	--

9.11.2 AbstractObservationCollectionType

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

Requirement /req/obs-core/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.
---	--

10 Basic Observations

10.1 General

10.1.1 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/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

10.1.2 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.
---	--

10.1.3 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.
---	---

10.2 Observation

10.2.1 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

Observation from the Basic Observations is described as a class diagram in Figure 17. The schema is fully described in 10.2.

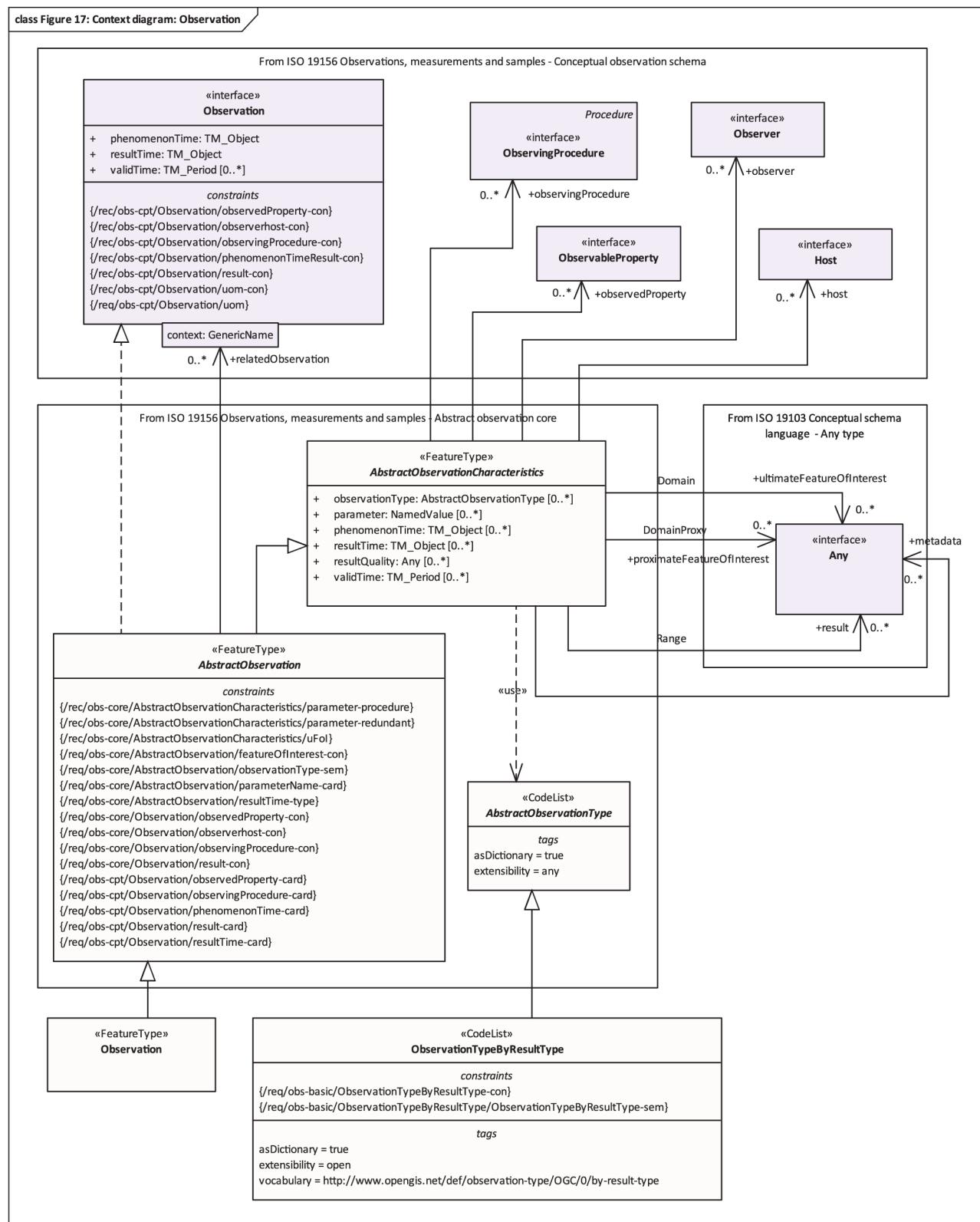


Figure 17 — Context diagram for Basic Observations — Observation

10.3 ObservationCharacteristics

10.3.1 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

ObservationCharacteristics from the Basic Observations is described as a class diagram in Figure 18. The schema is fully described in 10.3.

10.4 ObservationCollection

10.4.1 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
Imports	/req/obs-core/AbstractObservationCollection

ObservationCollection from the Basic Observations is described as a class diagram in Figure 18. The schema is fully described in 10.4.

10.5 ObservingCapability

10.5.1 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
-------------	--

ObservationCollection from the Basic Observations is described as a class diagram in Figure 18. The schema is fully described in 10.5.

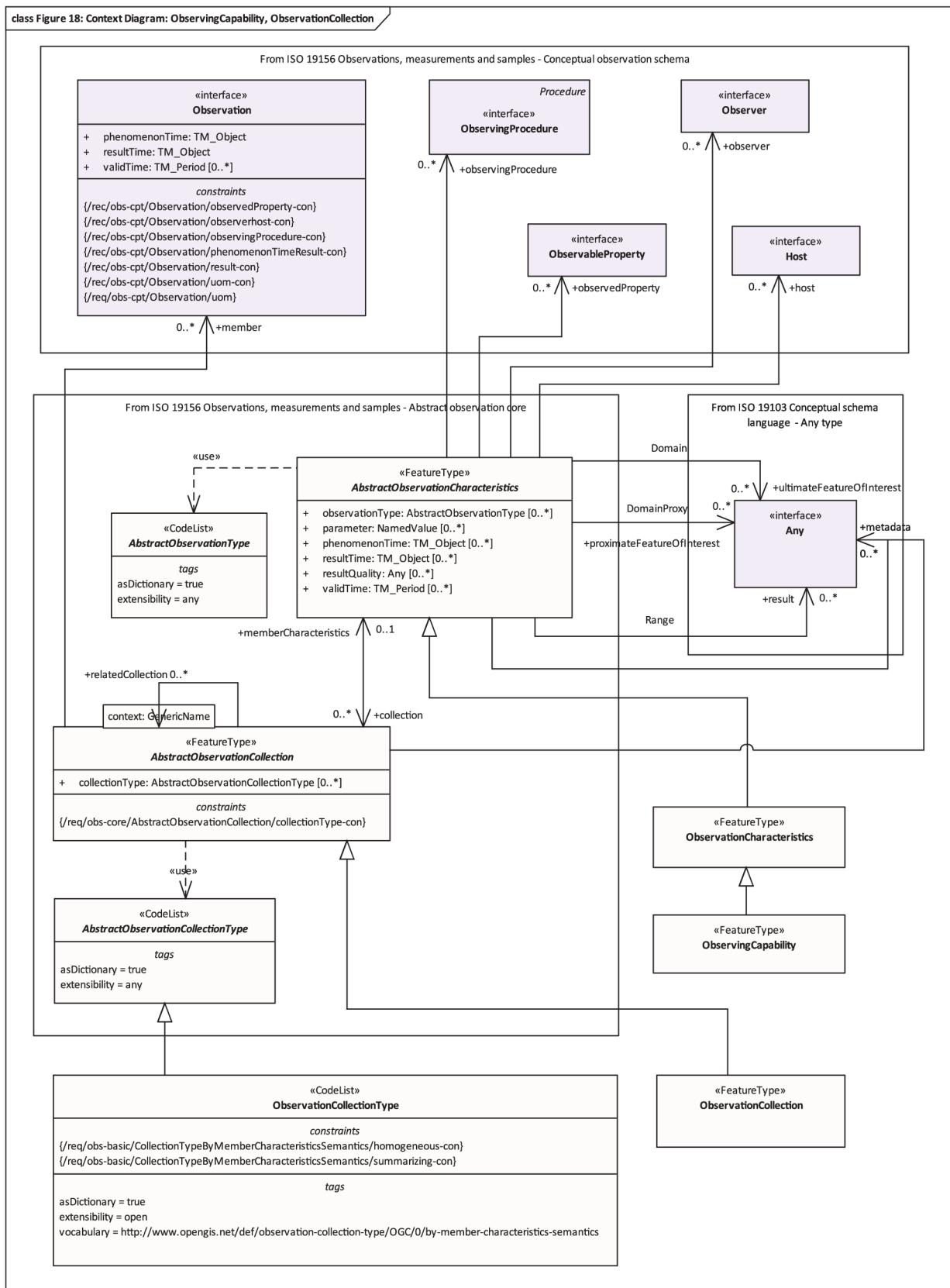


Figure 18 — Context diagram for Basic Observations — ObservationCharacteristics, ObservingCapability and ObservationCollection

10.5.2 Feature type ObservingCapability

Requirement <code>/req/obs-basic/ObservingCapability/ObservingCapability-sem</code>	An ObservingCapability shall be defined as information on Observation(s) that could potentially be provided.
---	--

EXAMPLE In order to explicitly describe the capabilities of an environmental monitoring facility, information on which Observable Properties are being measured with which methodology is provided.

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

- a) Some monitoring may have just one ObservingCapability:
 - 1) ObservingCapability:
 - i) ultimateFeatureOfInterest: 'Hydrogeological Unit 121AS';
 - ii) proximateFeatureOfInterest:'xyz';
 - iii) procedure: 'Groundwater depth measurement by electronic probe';
 - iv) observedProperty: 'GroundWaterDepth'.
- b) Other monitoring may have several such ObservingCapabilities, for example:
 - 1) ObservingCapability 1:
 - i) ultimateFeatureOfInterest: 'Entite hydrogeologique 143AE05';
 - ii) proximateFeatureOfInterest: 'Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE';
 - iii) procedure: 'Groundwater depth measurement by electronic probe';
 - iv) observedProperty: 'GroundWaterDepth'.
 - 2) ObservingCapability 2:
 - i) ultimateFeatureOfInterest: 'Entite hydrogeologique 143AE05';
 - ii) proximateFeatureOfInterest : 'Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE' ;
 - iii) procedure: 'Digital recording teletransmitted';
 - iv) observedProperty: 'Water Temperature'.
 - 3) ObservingCapability 3:
 - i) ultimateFeatureOfInterest: 'Entite hydrogeologique 143AE05';
 - ii) proximateFeatureOfInterest : 'Calcaires du Muschelkalk de Lorraine à SERVIGNY-LES-RAVILLE' ;

- iii) procedure: 'Digital recording teletransmitted';
- iv) observedProperty: 'Water conductivity measured at 25 °C'.

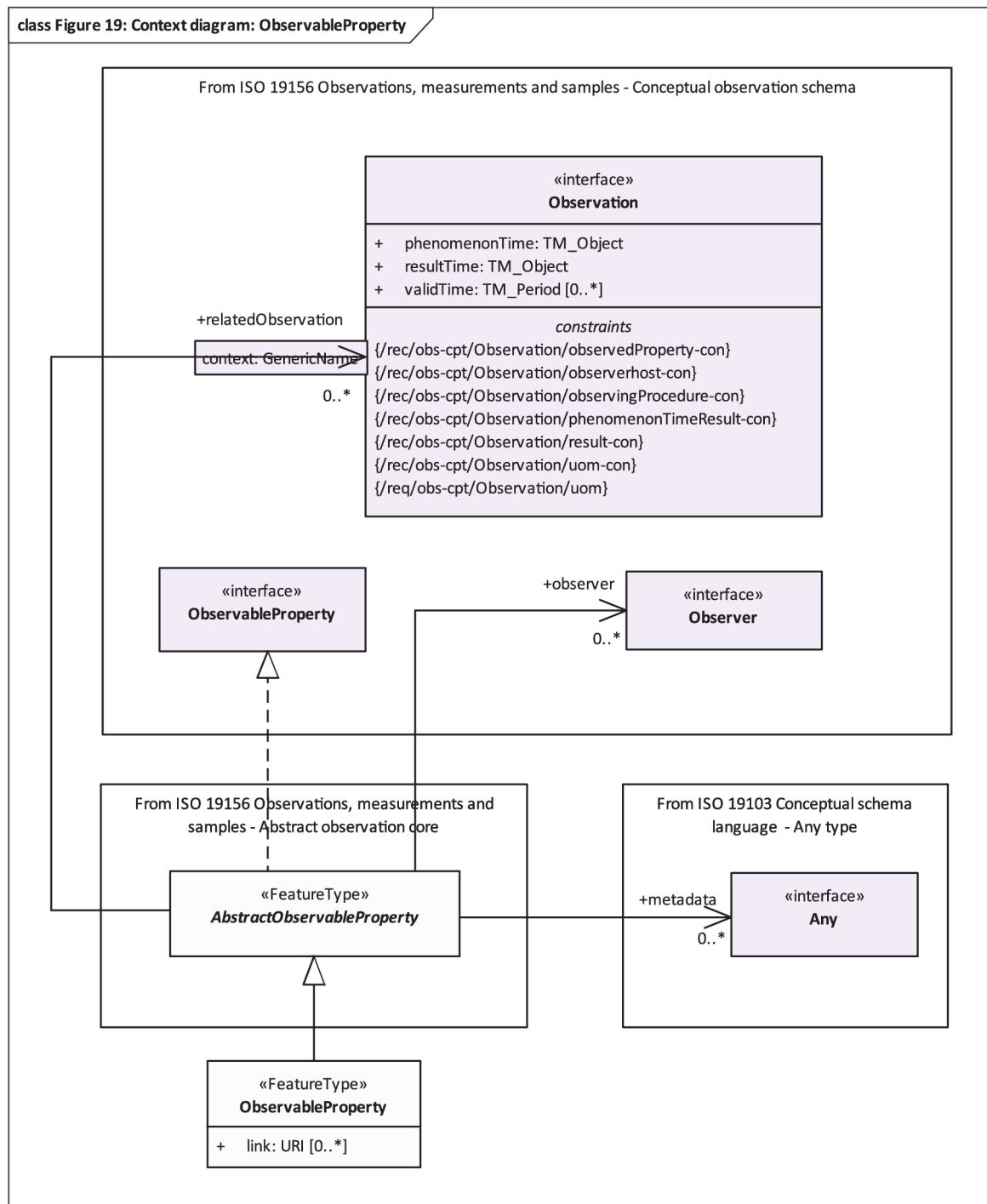
NOTE In the example above, URIs have been removed and only the labels provided for better readability.

10.6 ObservableProperty

10.6.1 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

ObservableProperty from the Basic Observations is described as a class diagram in Figure 19. The schema is fully described in 10.6.

**Figure 19— Context diagram for the Basic Observations — ObservableProperty**

10.7 ObservingProcedure

10.7.1 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

ObservingProcedure from the Basic Observations is described as a class diagram in Figure 20. The schema is fully described in 10.7.

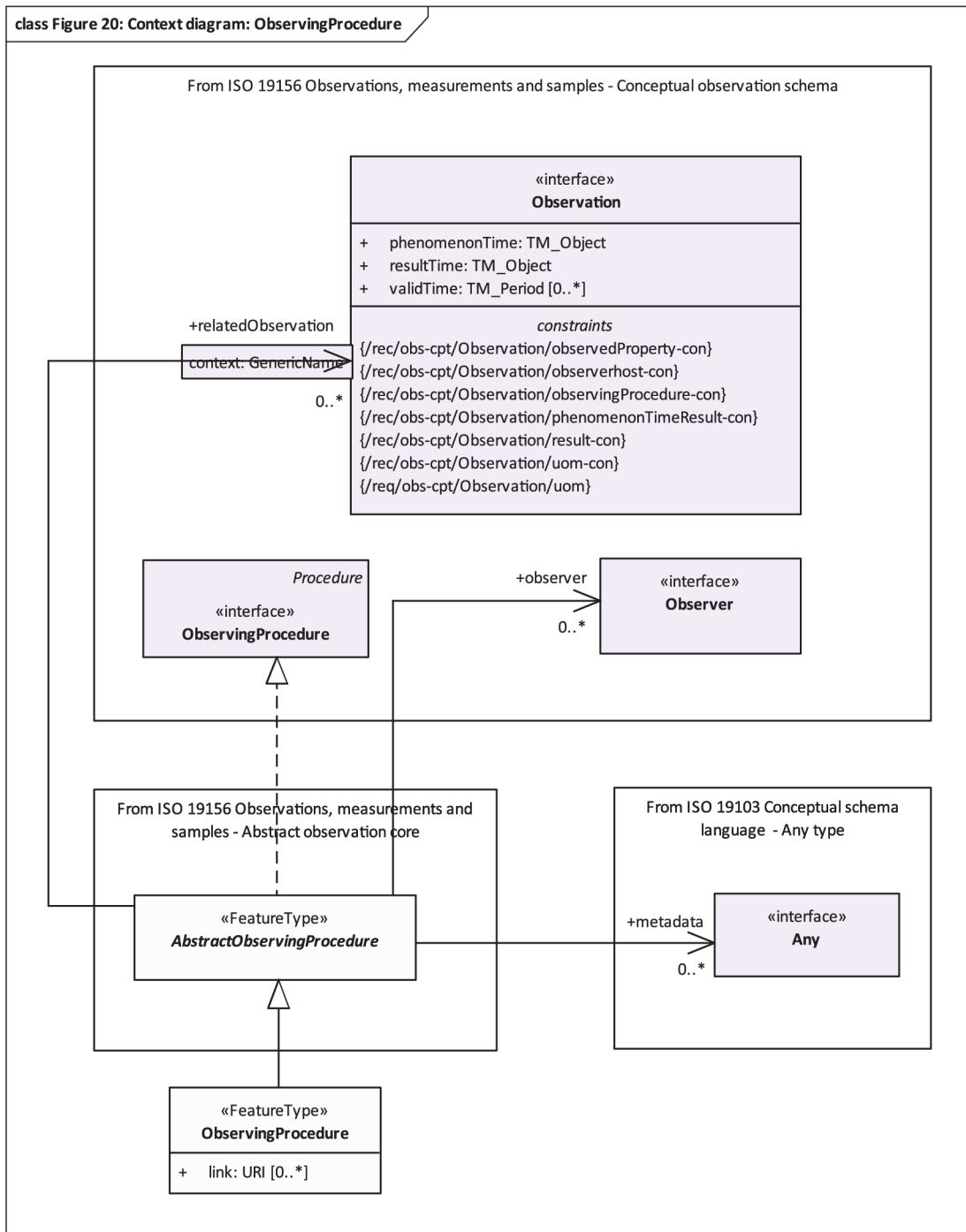


Figure 20 — Context diagram for Basic Observations — ObservingProcedure

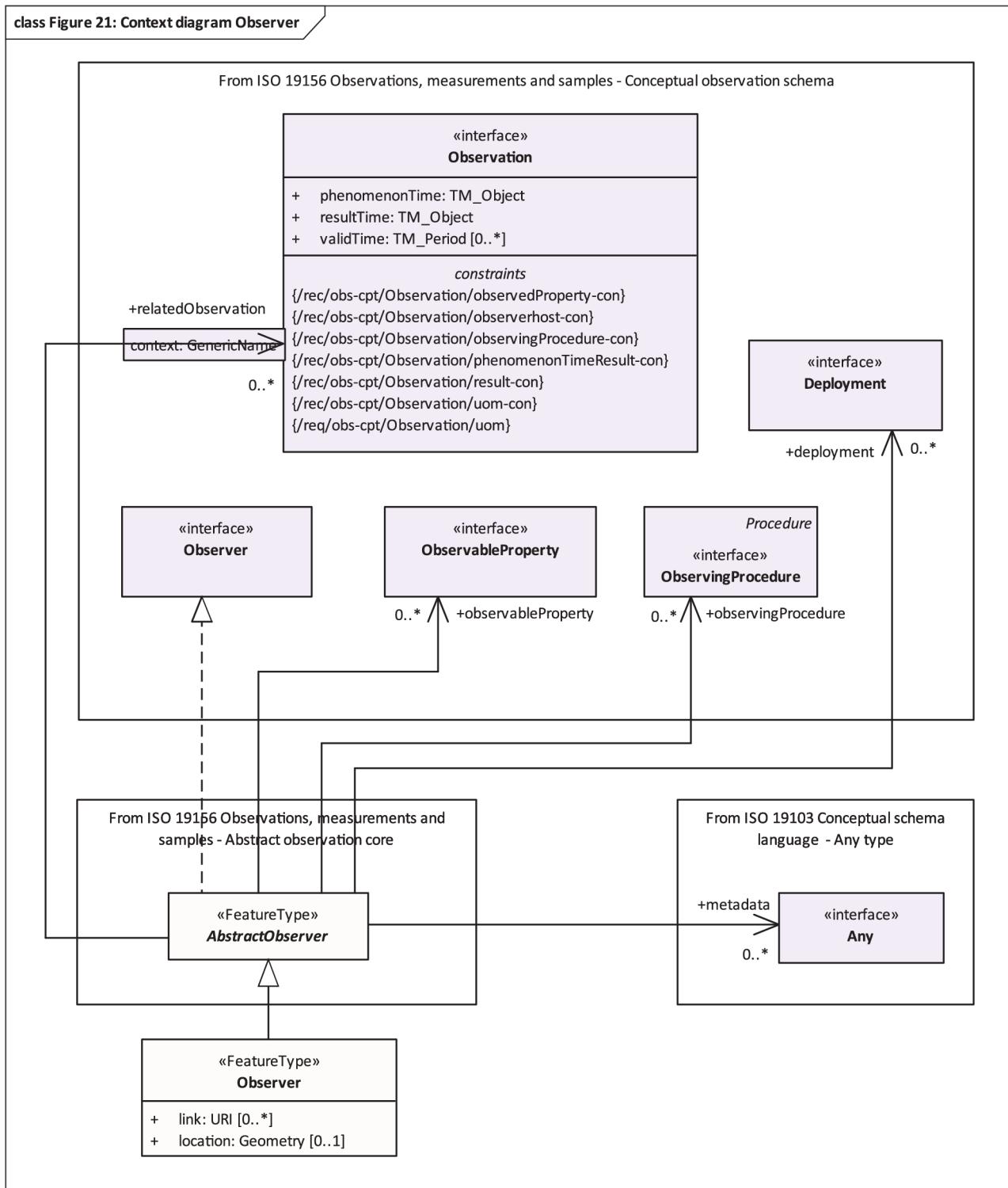
10.8 Observer

10.8.1 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

Observer from the Basic Observations is described as a class diagram in Figure 21. The schema is fully described in 10.8.

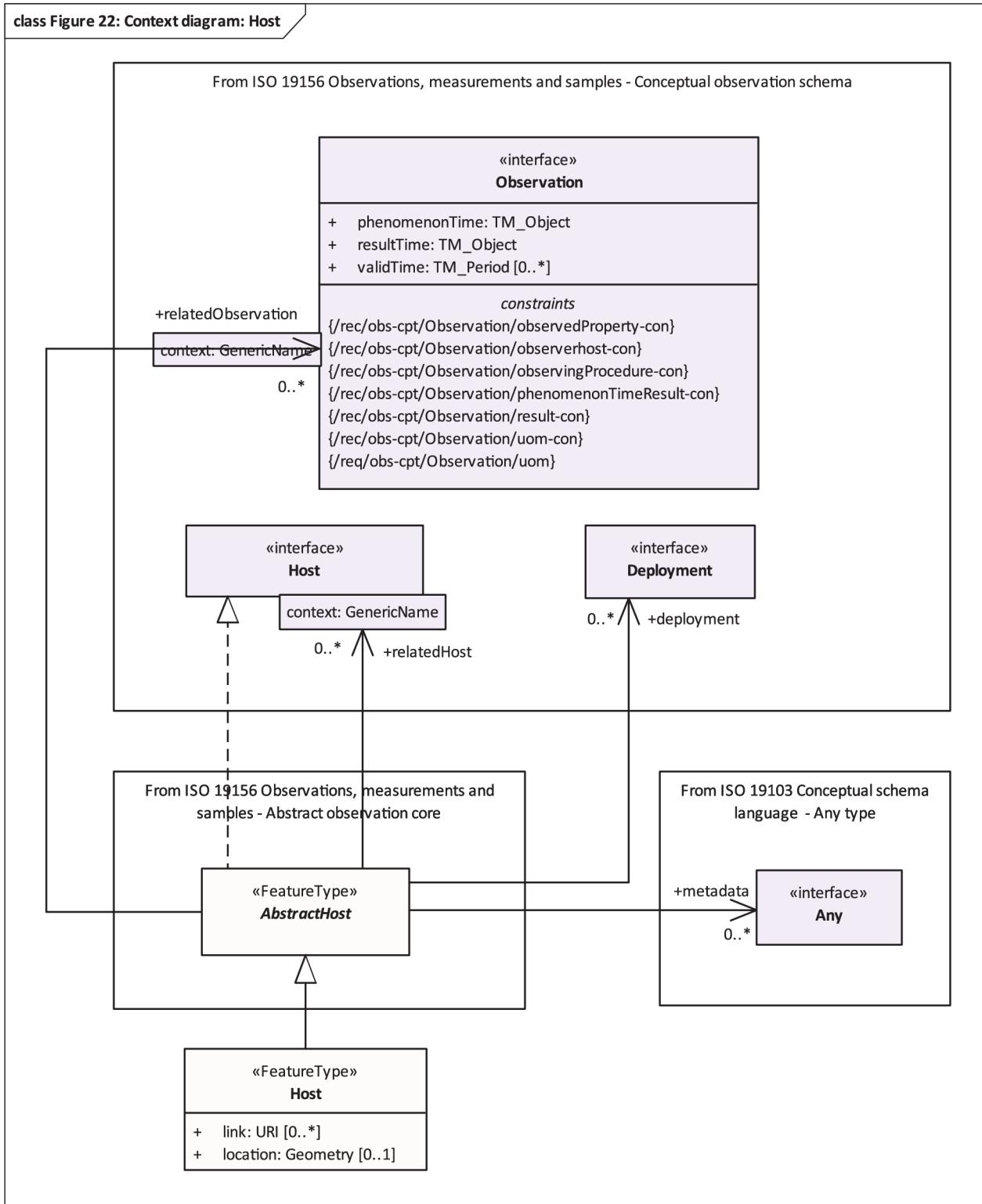
**Figure 21 — Context diagram for Basic Observations — Observer**

10.9 Host

10.9.1 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

Host from the Basic Observations is described as a class diagram in Figure 22. The schema is fully described in 10.9.

**Figure 22 — Context diagram for Basic Observations — Host**

10.10 Deployment

10.10.1 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

Deployment from the Basic Observations is described as a class diagram in Figure 23. The schema is fully described in 10.10.

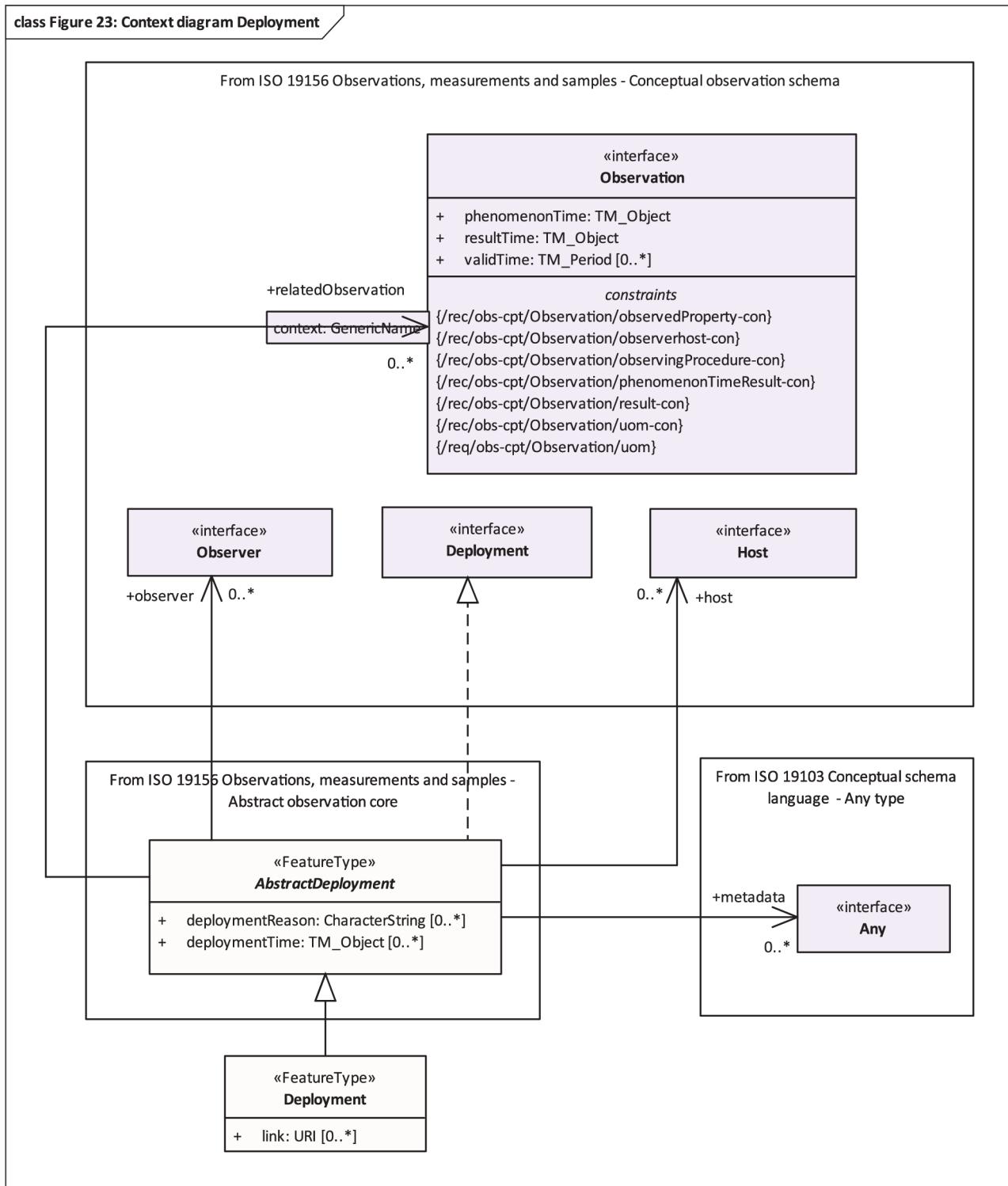


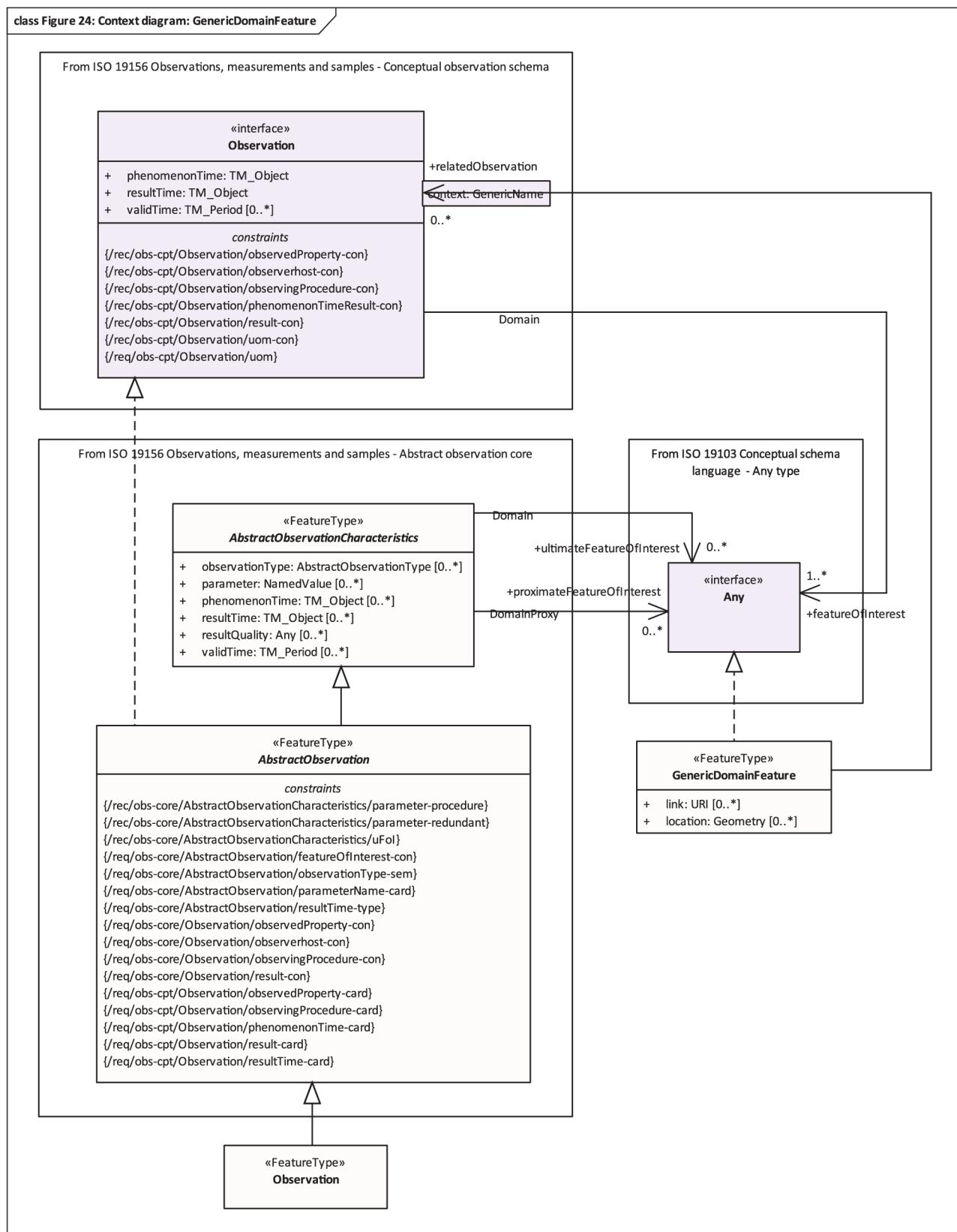
Figure 23 — Context diagram for Basic Observations — Deployment

10.11 GenericDomainFeature

10.11.1 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

GenericDomainFeature from the Basic Observations is described as a class diagram in Figure 24. The schema is fully described in 10.11.



NOTE GenericDomainFeature can be used as the target of the ultimate or proximate feature-of-interest of an Observation in the absence of an existing, more specific domain feature.

Figure 24 — Context diagram for Basic Observations — GenericDomainFeature

10.11.2 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.

10.12 Codelists

10.12.1 ObservationCollectionType

The codelist ObservationCollectionType realizes the AbstractObservationCollectionType and has the following values defined in this document: “homogeneous” and “summarizing”.

Requirement /req/obs-basic/ObservationCollectionType/ObservationCollectionType-sem	<p>The following entries shall be provided:</p> <ul style="list-style-type: none"> — homogeneous: all observations contained are of a similar nature; — summarizing: a wider grab-bag type of collection.
--	---

Requirement /req/obs-basic/ObservationCollectionType/homogeneous-con	<p>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.</p> <p>If a property value is provided within the ObservationCharacteristics, this value applies to all Observations contained in the ObservationCollection:</p> <ul style="list-style-type: none"> — 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 will refer to that ObservableProperty.

Requirement <pre>/req/obs- basic/ObservationCollectionType/summarizing- con</pre>	<p>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.</p> <p>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:</p> <ul style="list-style-type: none"> — 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 can 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:

- a) 1;
- b) 9;
- c) 2, 5 (a set with the two values);
- d) [8,1 -9,2] (a range of 8,1 to 9,2);
- e) 1, 2, 5 [8-11], (the exact set of values from the collection).

EXAMPLE 4 If the summarizing collection supplies:

- a) ultimateFeatureOfInterest=<https://example.org/collections/42/items/42>;
- b) deployment=[empty/NIL/null] (i.e. property provided but with no content);
- c) 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.

10.12.2 ObservationTypeByResultType

The codelist ObservationTypeByResultType is a specialization of AbstractObservationType created to support the legacy observation types from the previous version of this document.

	The following entries shall be provided:
--	--

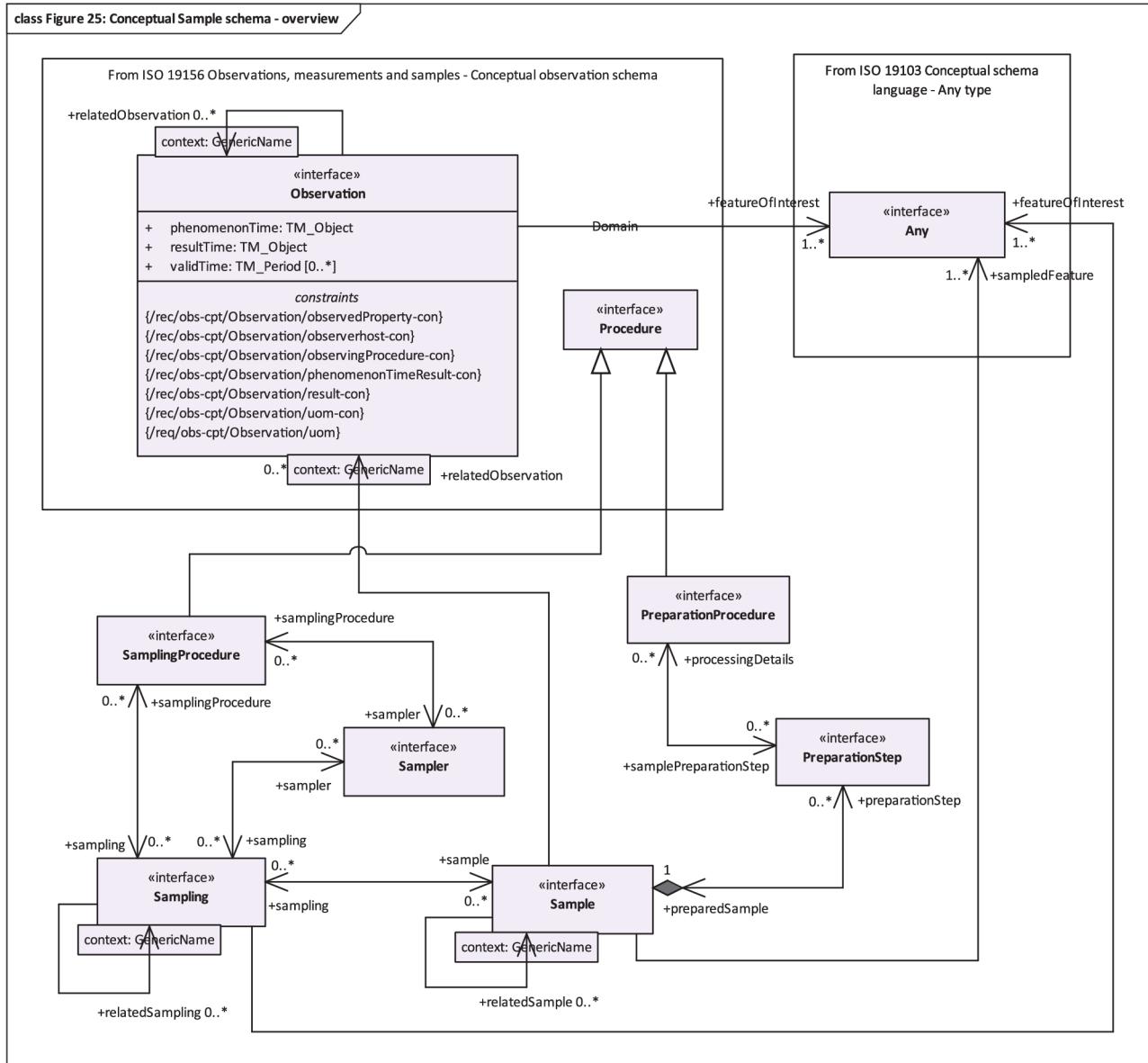
<p>Requirement</p> <p>/req/obs- basic/ObservationTypeByResultType/Observat ionTypeByResultType-sem</p>	<ul style="list-style-type: none"> — measurement: the result is of type Measure; — category-observation: the result is of type ScopedName; — truth-observation: result is a truth value; — count-observation: 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).
<p>Requirement</p> <p>/req/obs- basic/ObservationTypeByResultType/Observat ionTypeByResultType-con</p>	<p>The following constraints shall be applied to the value of the result association of the Observation based on the codelist value used:</p> <ul style="list-style-type: none"> — 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.

11 Conceptual Sample schema

11.1 General

11.1.1 Conceptual Sample schema model

The Conceptual Sample schema is described as a class diagram in Figure 25. It is fully described in 11.1.2.



NOTE A Sample can act as a proxy for the ultimate feature-of-interest of an Observation, and can be associated with this Observation by the role `featureOfInterest` as a specialization of `Any`. In this case the `sampledFeature` association of Sample would point upwards in the chain of sampled features leading to the ultimate feature-of-interest of the Observation. The Sample can associate itself with the Observation in question by the role `relatedObservation`.

Figure 25 — Conceptual Sample schema overview

11.1.2 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

11.2 Sample

11.2.1 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

Requirements Class	/req/sam-cpt/Sample
Requirement	/req/obs-cpt/gen/relatedObservation-sem

11.2.2 Interface Sample

Requirement /req/sam-cpt/Sample/Sample-sem	A Sample shall be defined as an object that is representative of a concept, real-world object or phenomenon.
--	---

NOTE 1 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).

NOTE 2 The physical characteristics of the features themselves are of little interest, except perhaps to the manager of a sampling campaign.

NOTE 3 Typically, the Sample is a Feature which is intended to be representative of a FeatureOfInterest on which Observations can be made. As such, it can 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.

11.2.3 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 .
--	---

11.2.4 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 .
---	--

11.2.5 Association sampledFeature

Requirement /req/sam-cpt/Sample/sampledFeature-sem	The sampledFeature is the feature the 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.

11.2.6 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.

11.3 Sampling

11.3.1 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

11.3.2 Interface Sampling

Requirement /req/sam-cpt/Sampling/Sampling-sem	A Sampling shall be defined as an act applying a SamplingProcedure to create or transform one or more Sample(s) .
--	--

EXAMPLE 1 Crushing a rock sample in a ball mill.

EXAMPLE 2 Digging a pit through a soil sequence.

EXAMPLE 3 Dividing a field site into quadrants.

EXAMPLE 4 Drawing blood from a patient.

EXAMPLE 5 Extracting water from an observation well.

EXAMPLE 6 Extracting a sample from a defined environmental monitoring station.

EXAMPLE 7 Registering an image of the landscape.

EXAMPLE 8 Sieving a powder to separate the subset finer than 100-mesh.

EXAMPLE 9 Selecting a subset of a population.

EXAMPLE 10 Splitting a piece of drill-core to create two new samples.

EXAMPLE 11 Taking a diamond-drill core from a rock outcrop.

11.3.3 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 .
--	---

11.3.4 Association featureOfInterest

Requirement /req/sam-cpt/Sampling/featureOfInterest-sem	A feature-of-interest shall be defined as the concept, real-world object or phenomenon (feature-of-interest) the Sample(s) of the Sampling represent. Reference to the feature-of-interest shall be done using the association with the role featureOfInterest .
---	---

Requirement /req/sam-cpt/Sampling/featureOfInterest-card	A Sampling shall have at least 1 featureOfInterest and may have more than 1 in cases where multiple objects are sampled with the intention to create one Sample .
--	--

11.3.5 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 .
--	---

11.3.6 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 .
---	--

11.3.7 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.
---	---

11.4 Sampler

11.4.1 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

11.4.2 Interface Sampler

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

EXAMPLE 1 A ball mill, diamond drill, hammer.

EXAMPLE 2 A hypodermic syringe and needle.

EXAMPLE 3 An image sensor, a soil auger.

EXAMPLE 4 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 is not necessarily a physical device.

11.4.3 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 .
---	--

11.4.4 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 .
---	--

11.5 PreparationStep

11.5.1 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

11.5.2 Interface PreparationStep

Requirement /req/sam-cpt/PreparationStep/PreparationStep-sem	A PreparationStep shall be defined as an individual step pertaining to a PreparationProcedure .
--	---

11.5.3 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 .
--	---

11.5.4 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 .
---	--

11.6 PreparationProcedure

11.6.1 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

11.6.2 Interface PreparationProcedure

Requirement /req/sam-cpt/PreparationProcedure/PreparationProcedure-sem	A PreparationProcedure shall be defined as the description of preparation steps performed on a Sample .
--	---

11.6.3 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.
---	---

11.7 SamplingProcedure

11.7.1 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

11.7.2 Interface SamplingProcedure

Requirement /req/sam-cpt/SamplingProcedure/SamplingProcedure-sem	A SamplingProcedure shall be defined as the description of steps performed by a Sampler in order to extract a Sample from its sampledFeature in the frame of a Sampling .
--	--

11.7.3 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.
---	--

11.7.4 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.
--	--

12 Abstract Sample Core

12.1 General

12.1.1 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

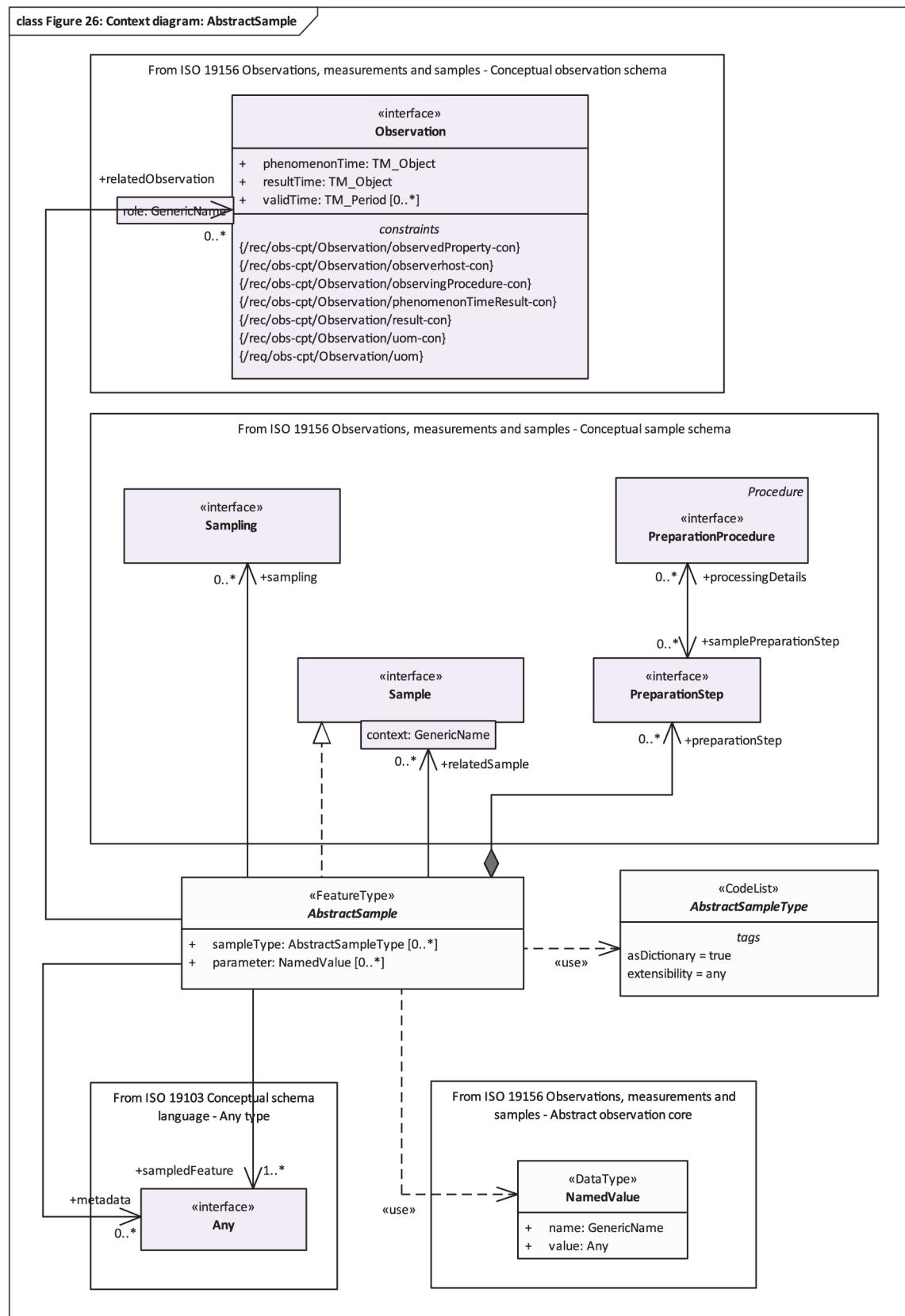
12.2 AbstractSample

12.2.1 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

AbstractSample from the Abstract Sample Core is described as a class diagram in Figure 26. The schema is fully described in 12.2.

**Figure 26 — Context diagram for Abstract Sample Core — AbstractSample**

12.2.2 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.
---	---

12.2.3 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 needs to 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 Using the classes, attributes and associations explicitly modelled in the OMS greatly improves the interoperability compared to using the generic parameter mechanism to include the same information.

12.3 AbstractSampling

12.3.1 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

Requirements Class	/req/sam-core/AbstractSampling
Requirement	/req/obs-core/gen/metadata-sem

AbstractSampling from the Abstract Sample Core is described as a class diagram in Figure 27. The schema is fully described in 12.3.

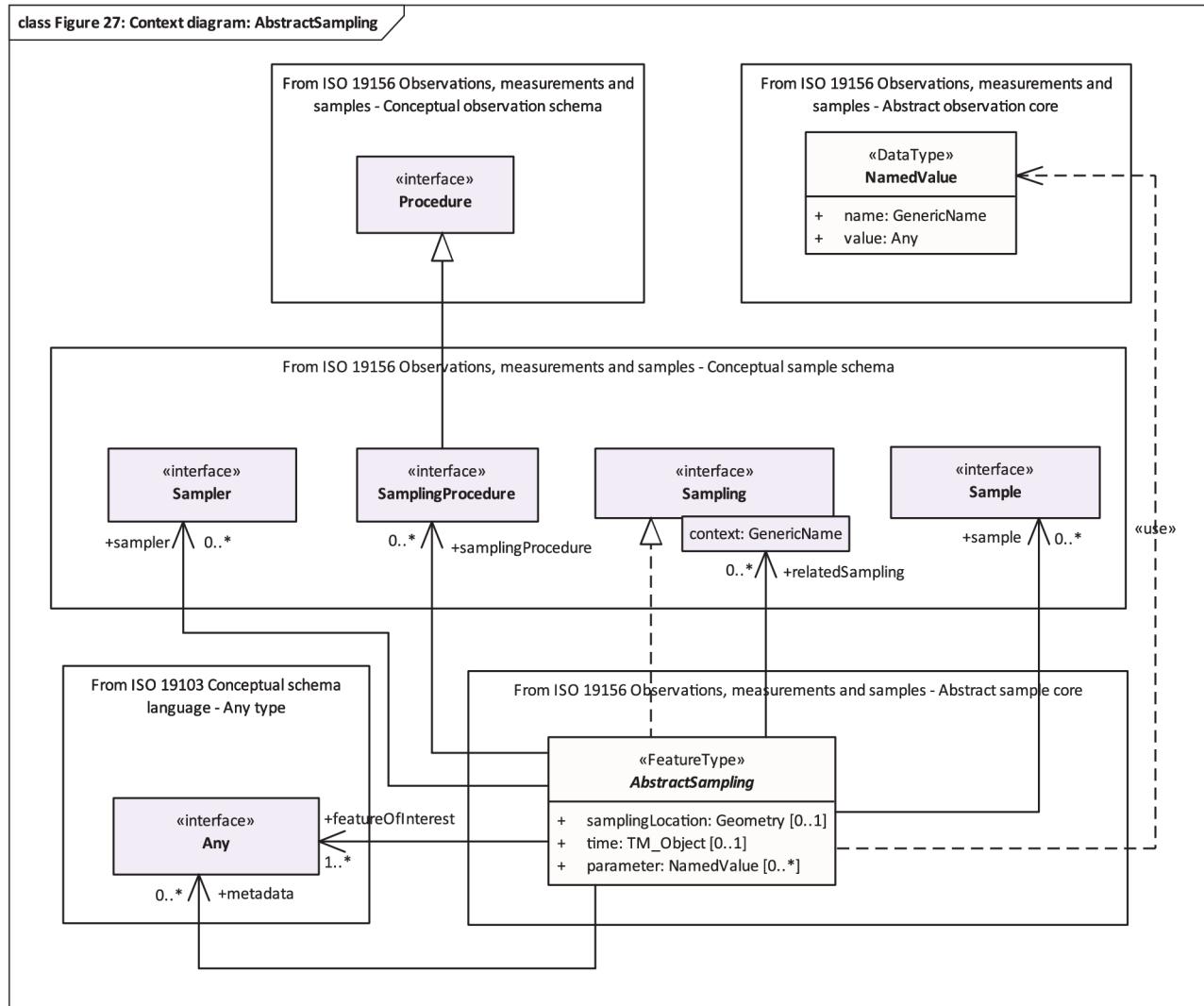


Figure 27 — Context diagram for Abstract Sample Core — AbstractSampling

12.3.2 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.
---	--

12.3.3 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.
---	---

12.3.4 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 needs to pass to allow sediments to settle. The exact waiting time used in this Sampling can be stored in the parameter.

NOTE Using the classes, attributes and associations explicitly modelled in the OMS greatly improves the interoperability compared to using the generic parameter mechanism to include the same information.

12.4 AbstractSampler

12.4.1 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

AbstractSampler from the Abstract Sample Core is described as a class diagram in Figure 28. The schema is fully described in 12.4.

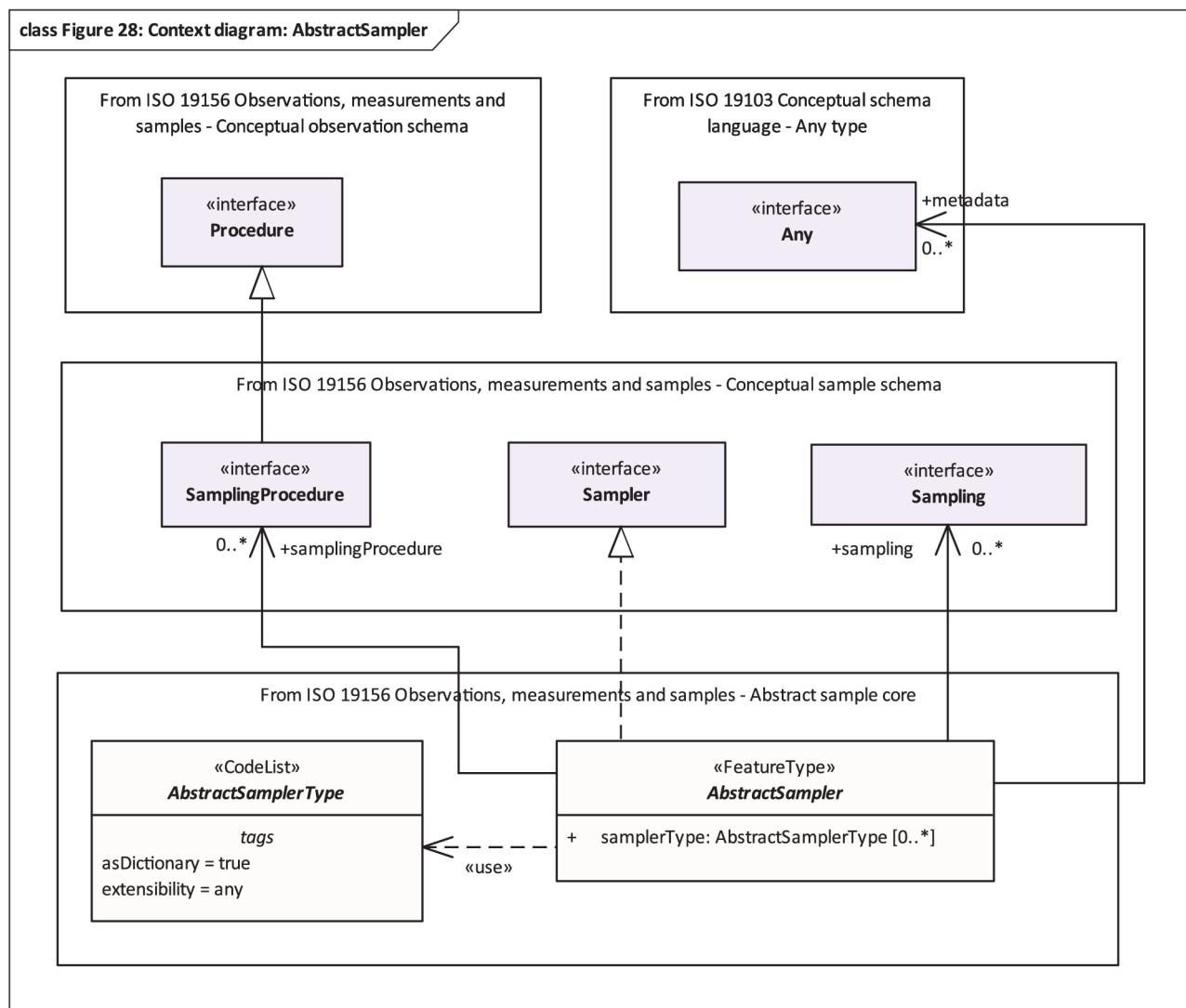


Figure 28 — Context diagram for the Abstract Sample Core — AbstractSampler

12.4.2 Attribute samplerType

Requirement <code>/req/sam-core/AbstractSampler/samplerType-sem</code>	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.
--	---

EXAMPLE 1 A ball mill, diamond drill, hammer.

EXAMPLE 2 A hypodermic syringe and needle.

EXAMPLE 3 An image sensor, a soil auger.

EXAMPLE 4 A human being.

12.5 AbstractSamplingProcedure

12.5.1 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

AbstractSamplingProcedure from the Abstract Sample Core is described as a class diagram in Figure 29. The schema is fully described in 12.5.

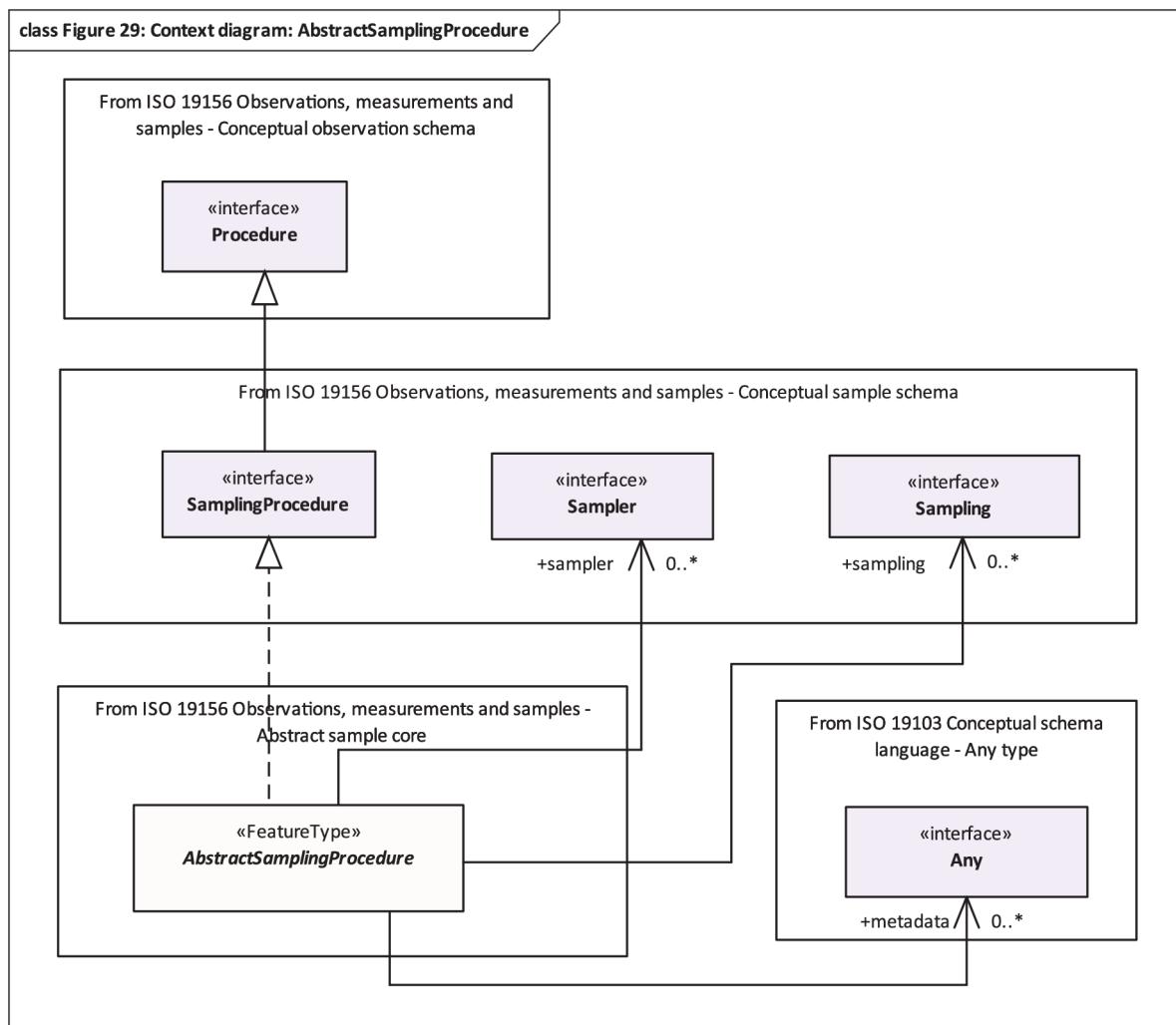


Figure 29 — Context diagram for Abstract Sample Core — AbstractSamplingProcedure

12.6 AbstractPreparationProcedure

12.6.1 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

AbstractPreparationProcedure and AbstractPreparationStep from the Abstract Sample Core are described as a class diagram in Figure 30. The schema is fully described in 12.6.

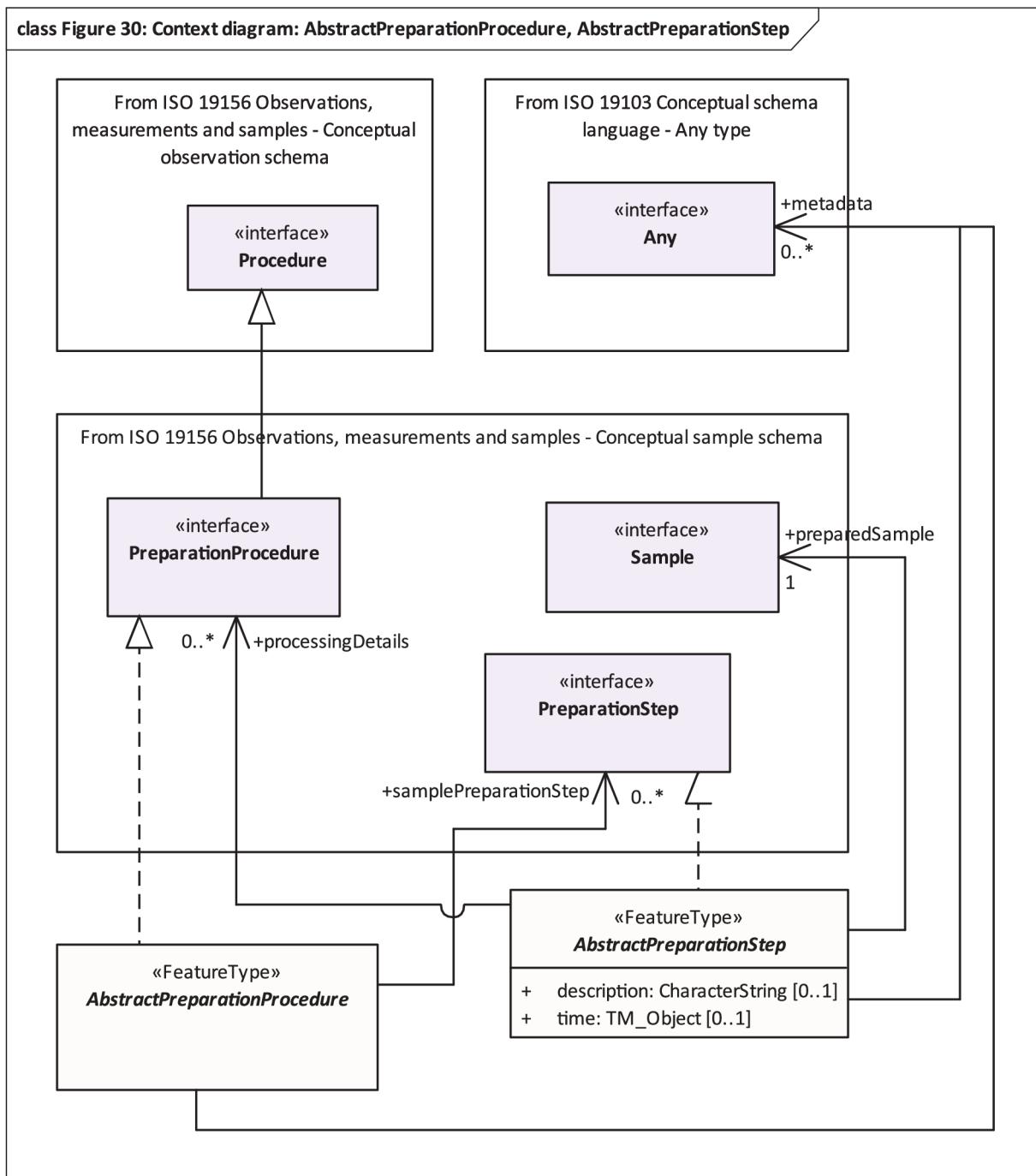


Figure 30 — Context diagram for Abstract Sample Core —AbstractPreparationProcedure and AbstractPreparationStep

12.7 AbstractPreparationStep

12.7.1 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

12.7.2 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.
---	---

12.7.3 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.
--	---

12.8 Codelists

12.8.1 AbstractSampleType

The codelist AbstractSampleType can be specialized as required to more precisely define the 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.
---	--

12.8.2 AbstractSamplerType

The codelist AbstractSamplerType can be specialized as required to more precisely define the 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.
---	--

13 Basic Samples

13.1 General

13.1.1 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

13.2 Sample

13.2.1 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

Sample, SpatialSample, StatisticalSample and MaterialSample from the Basic Samples are described as a class diagram in Figure 31. The schema is fully described in 13.2, 13.3, 13.4 and 13.5.

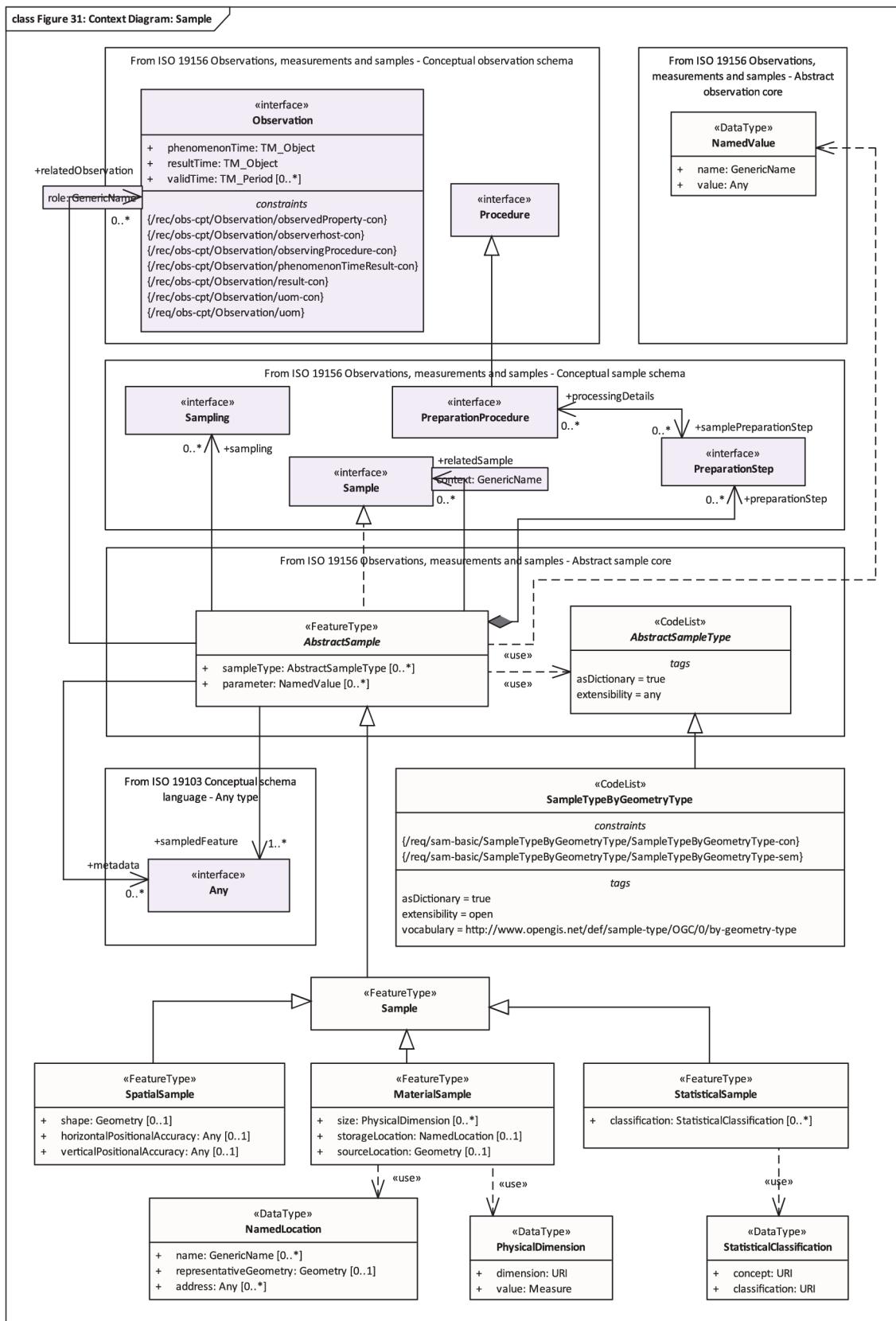


Figure 31— Context diagram for Basic Samples — Sample, SpatialSample, StatisticalSample and MaterialSample

13.3 SpatialSample

13.3.1 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

13.3.2 Feature type SpatialSample

Requirement /req/sam-basic/SpatialSample/SpatialSample-sem	A SpatialSample shall be defined as 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 can be extensive in one, two or three spatial dimensions.

EXAMPLE 1 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.

EXAMPLE 2 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.

13.3.3 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 can use different intervals, and sub-samples can 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.

13.3.4 Attribute horizontalPositionalAccuracy

Requirement /req/sam-basic/SpatialSample/horizontalPositionalAccuracy-sem	The positional accuracy of the horizontal component of the Geometry of the SpatialSample . If horizontal positional accuracy information pertaining to the SpatialSample is provided, the attribute horizontalPositionalAccuracy:Any shall be used.
---	--

13.3.5 Attribute verticalPositionalAccuracy

Requirement /req/sam-basic/SpatialSample/verticalPositionalAccuracy-sem	The positional accuracy of the vertical component of the Geometry of the SpatialSample . If horizontal positional accuracy information pertaining to the SpatialSample is provided, the attribute verticalPositionalAccuracy:Any shall be used.
---	--

13.4 MaterialSample

13.4.1 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

Requirements Class	/req/sam-basic/MaterialSample
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

13.4.2 Feature type MaterialSample

Requirement /req/sam-basic/MaterialSample/MaterialSample-sem	A MaterialSample shall be defined as 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.

13.4.3 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 can be length, mass, volume, etc. as appropriate for the MaterialSample instance and its material type.

13.4.4 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 can be a location such as a shelf in a warehouse or a drawer in a museum.

13.4.5 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.
--	--

NOTE 1 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.

NOTE 2 The attribute sourceLocation of the MaterialSample can be unnecessary in the case that the related Sampling act samplingLocation attribute is provided.

13.5 StatisticalSample

13.5.1 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

13.5.2 Feature type StatisticalSample

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

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

EXAMPLE The male or female subset of a population.

13.5.3 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 can be age, gender, etc. as appropriate for the set or population on which the subsetting is performed.

13.6 Sampling

13.6.1 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
Requirement	/req/obs-basic/gen/link-sem

Sampling from the Basic Samples is described as a class diagram in Figure 32.

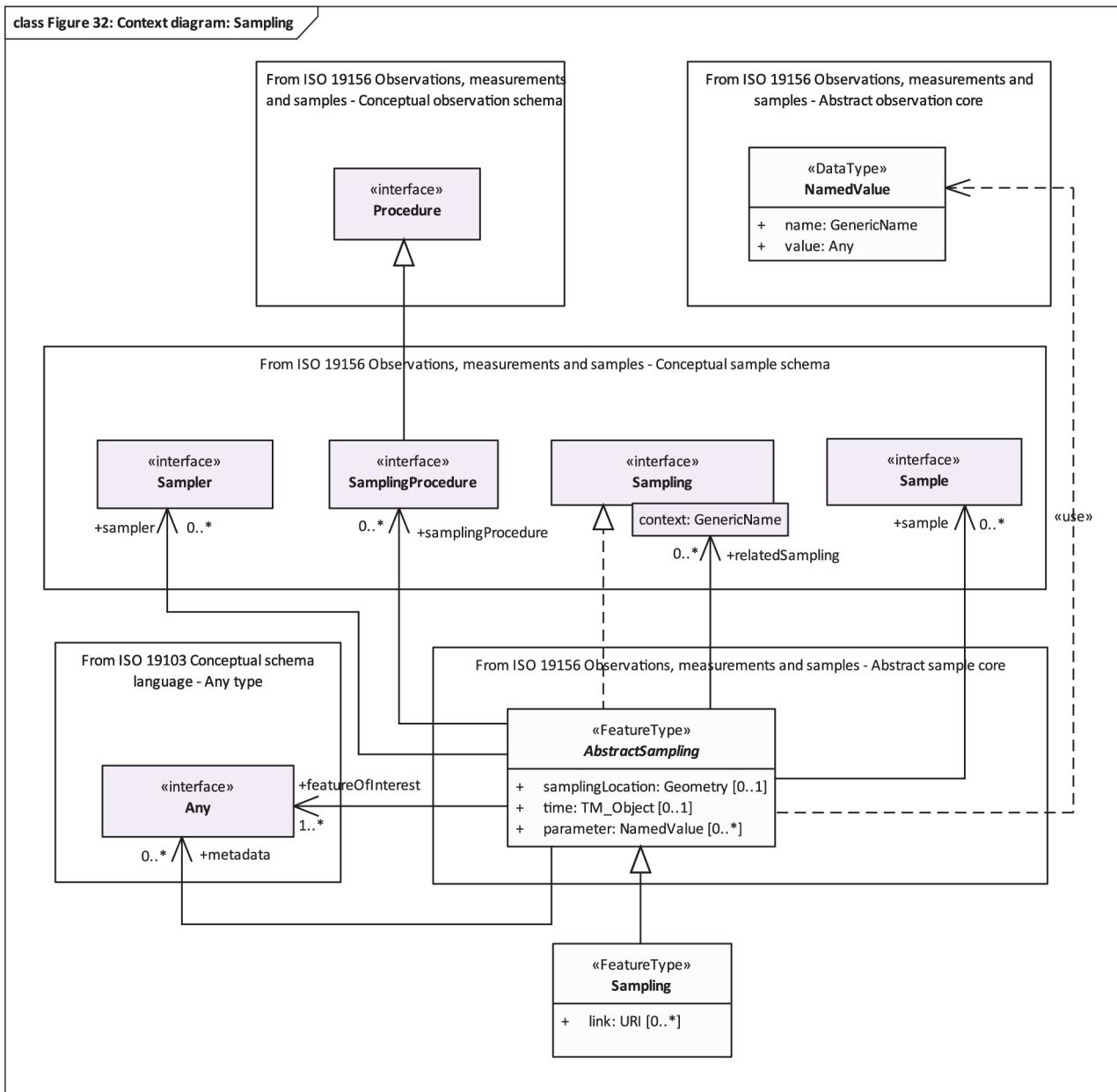


Figure 32 — Context diagram for Basic Samples — Sampling

13.7 Sampler

13.7.1 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
Requirement	/req/obs-basic/gen/link-sem

Sampler from the Basic Samples is described as a class diagram in Figure 33.

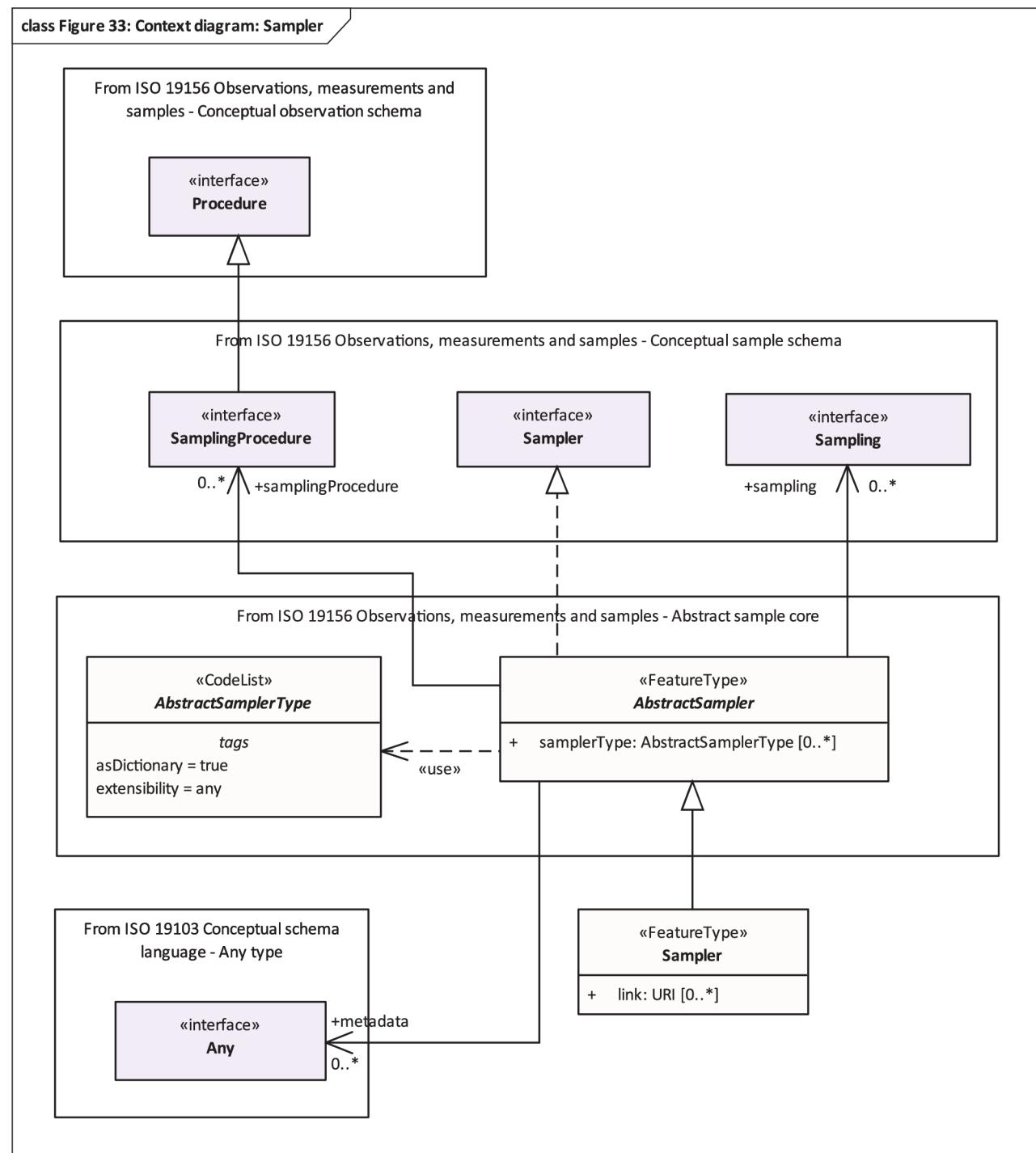


Figure 33 — Context diagram for Basic Samples — Sampler

13.8 SamplingProcedure

13.8.1 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

SamplingProcedure from the Basic Samples is described as a class diagram in Figure 34.

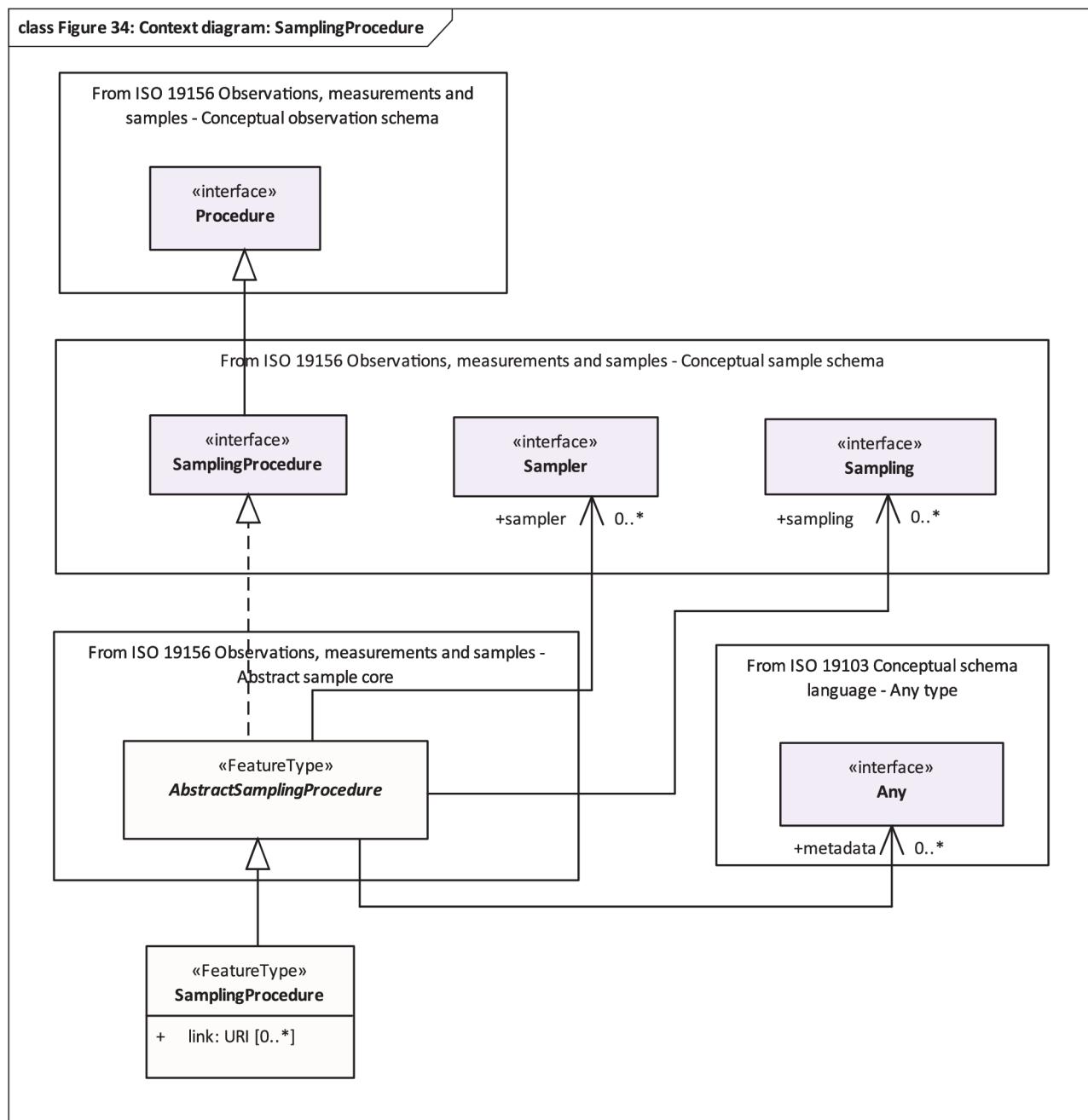


Figure 34—Context diagram for Basic Samples — SamplingProcedure

13.9 PreparationProcedure

13.9.1 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

PreparationProcedure from the Basic Samples is described as a class diagram in Figure 35.

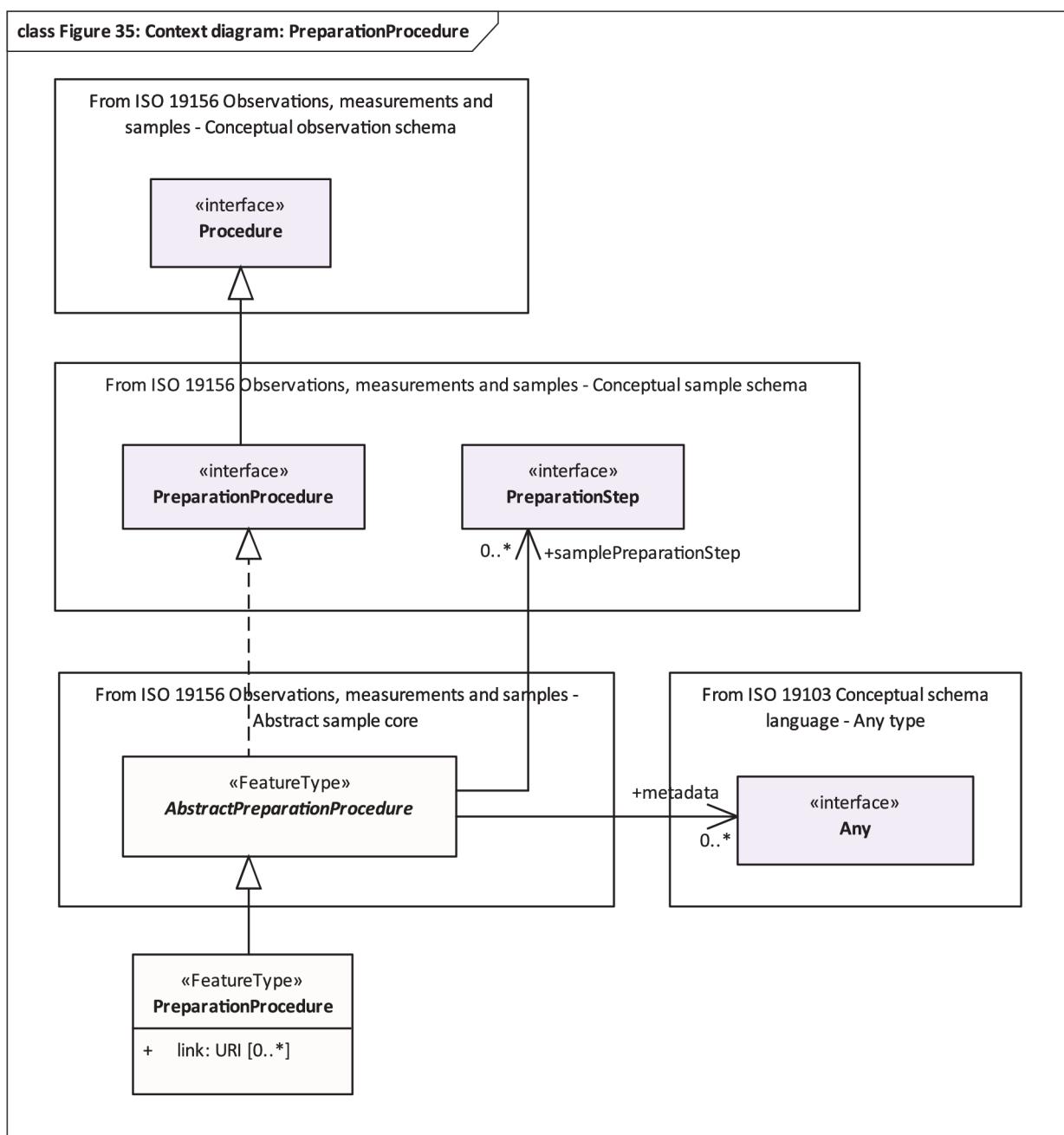


Figure 35 — Context diagram for Basic Samples — PreparationProcedure

13.10 PreparationStep

13.10.1 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

PreparationStep from the Basic Samples is described as a class diagram in Figure 36.

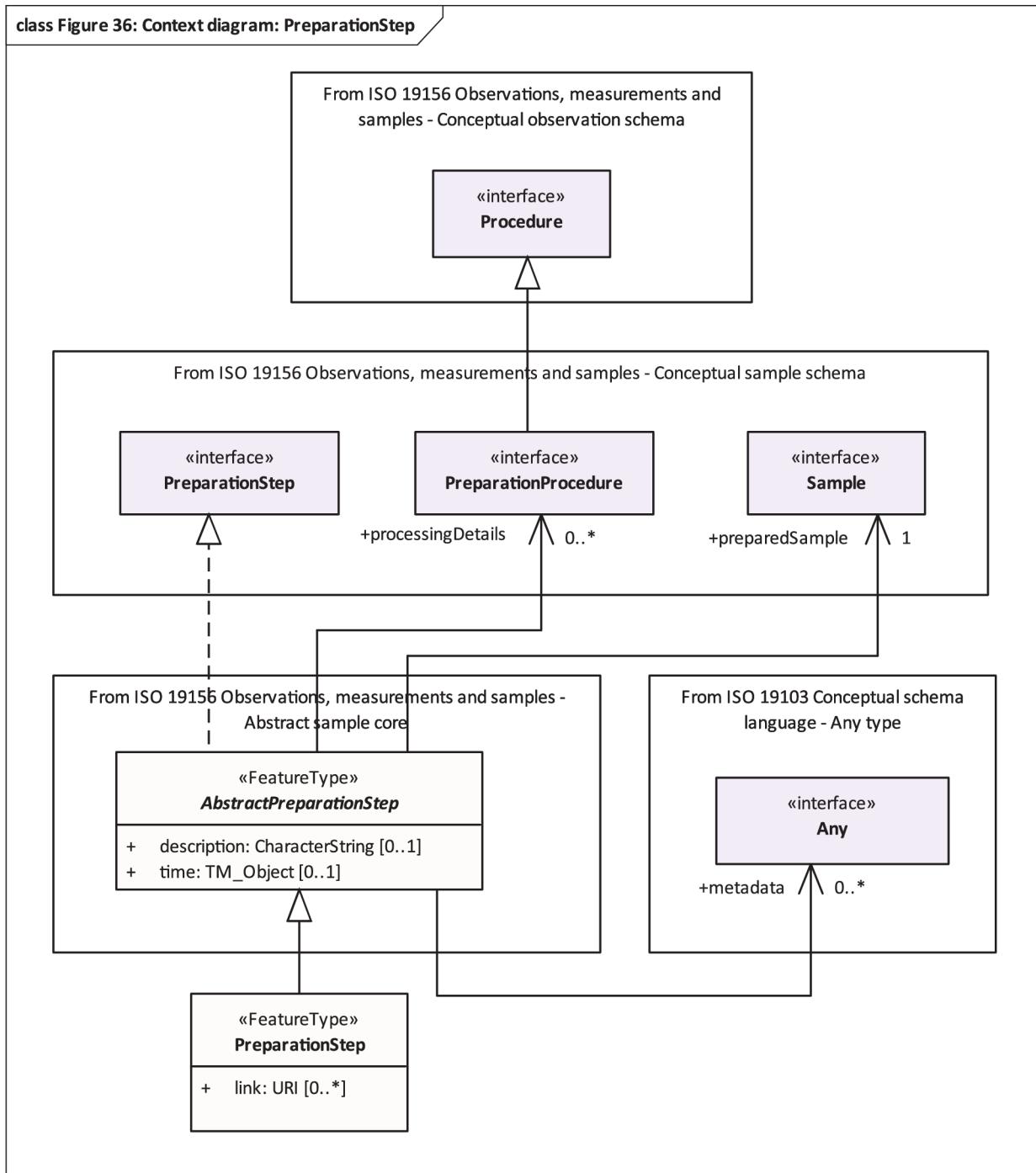


Figure 36 — Context diagram for Basic Samples — PreparationStep

13.11 SampleCollection

13.11.1 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

SampleCollection from the Basic Samples is described as a class diagram in Figure 37. The schema is fully described in 13.11.2, 13.11.3 and 13.11.4.

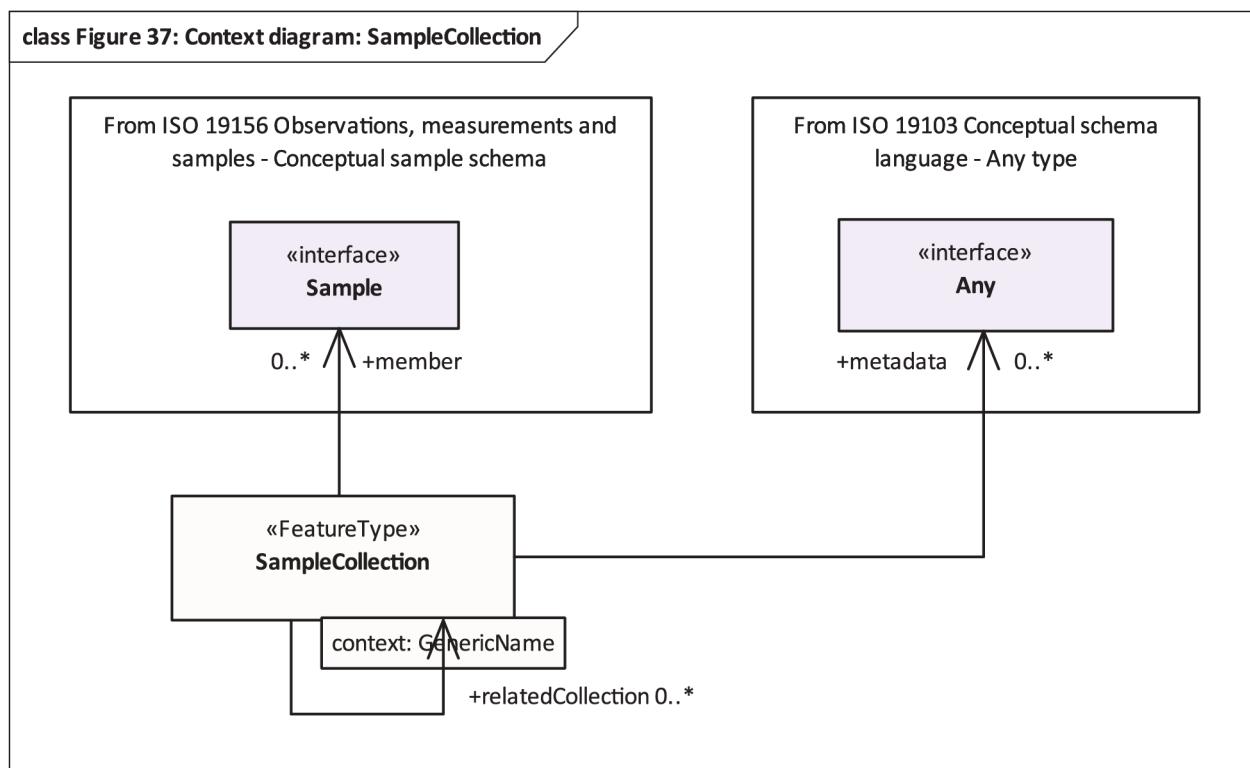


Figure 37 — Context diagram for Basic Samples — SampleCollection

13.11.2 Feature type SampleCollection

Requirement /req/sam-basic/SampleCollection/SampleCollection-sem	A SampleCollection shall be defined as a collection of Samples .
---	--

13.11.3 Association member

Requirement /req/sam-basic/SampleCollection/member-sem	A Sample that is part of this SampleCollection . If the SampleCollection has members, the association with the role member shall be used.
--	---

13.11.4 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.
---	---

13.12 PhysicalDimension

13.12.1 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

13.12.2 Data type PhysicalDimension

Requirement /req/sam-basic/PhysicalDimension/PhysicalDimension-sem	A PhysicalDimension shall be defined as a dataType for the provision of various size quantities.
--	---

13.12.3 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 .
--	--

13.12.4 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
--	--

13.13 NamedLocation

13.13.1 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

13.13.2 Data type NamedLocation

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

13.13.3 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.
--	---

13.13.4 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.
---	---

13.13.5 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.
---	--

13.14 StatisticalClassification

13.14.1 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

13.14.2 Data type StatisticalClassification

Requirement /req/sam-basic/StatisticalClassification/StatisticalClassification-sem	A StatisticalClassification shall be defined as a dataType for the provision of information on statistical classifications.
--	--

13.14.3 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.

13.14.4 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 1 The classification for a statistical classification could be:

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

13.15 Codelists

13.15.1 SampleTypeByGeometryType

The codelist SampleTypeByGeometryType is a specialization of AbstractSampleType created to support the legacy sample types from ISO 19156:2011.

Requirement /req/sam-basic/SampleTypeByGeometryType/SampleTypeByGeometryType-sem	The following entries shall be provided: <ul style="list-style-type: none"> — 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: <ul style="list-style-type: none"> — 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.
--	--

Annex A

(normative)

Abstract test suite

A.1 Abstract tests for Conceptual Observation schema package

A.1.1 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

A.1.2 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

A.1.3 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

A.1.4 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
-----------	------------

A.1.5 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

A.1.6 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

A.1.7 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

A.1.8 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

A.2 Abstract tests for Abstract Observation Core package

A.2.1 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

A.2.2 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

A.2.3 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

A.2.4 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

A.2.5 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

A.2.6 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

A.2.7 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

A.2.8 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

A.2.9 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

A.2.10 Abstract Observation Core – AbstractObservationCollection

Conformance class	/conf/obs-core/AbstractObservationCollection
Requirements	/req/obs-core/AbstractObservationCollection
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

A.3 Abstract tests for Basic Observations package

A.3.1 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

A.3.2 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

A.3.3 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

A.3.4 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

A.3.5 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

A.3.6 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

A.3.7 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

A.3.8 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

A.3.9 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

A.3.10 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

A.3.11 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

A.4 Abstract tests for Conceptual Sample schema package

A.4.1 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

A.4.2 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

A.4.3 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

A.4.4 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

A.4.5 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

A.4.6 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

A.4.7 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

A.5 Abstract tests for Abstract Sample Core package

A.5.1 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

A.5.2 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

A.5.3 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

A.5.4 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

A.5.5 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

A.5.6 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

A.5.7 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

A.6 Abstract tests for Basic Samples package

A.6.1 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

A.6.2 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

A.6.3 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

A.6.4 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

A.6.5 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

A.6.6 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

A.6.7 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

A.6.8 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

A.6.9 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

A.6.10 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

A.6.11 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

Annex B (informative)

Common usage of OMS concepts

B.1 Introduction

This document 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 document 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 OMS concepts to those used within some domains.

For the following domain, information has been provided in the tables listed:

- Earth Observations (EO): Table B.1
- Metrology: Table B.2
- Earth science simulations: Table B.3
- Assay/Chemistry: Table B.4
- Geology field observations: Table B.5
- Geotechnics observations: Table B.6
- Water quality observations: Table B.7
- Soil quality observations: Table B.8

B.2 Earth observations (EO)

Table B.1 — Earth observations (EO)

OMS	EO	Example
Observation::result	Observation value, measurement value, observation	35 µg/m ³
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

OMS	EO	Example
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

B.3 Metrology

Table B.2 — Metrology

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

B.4 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

B.5 Assay/Chemistry

Table B.4 — Assay/Chemistry

OMS	Geochemistry
Observation::proximateFeatureOfInterest:MaterialSample	Sample
MaterialSample::sampledFeature:GeologicUnit	Ore body, Geologic Unit

OMS	Geochemistry
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

B.6 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

B.7 Geotechnics observations

Table B.6 — Geotechnics observations

OMS	Geotechnical in situ test
Observation::result	A log
Observation::observedProperty	A soil property (e.g. Gamma ray, resistivity, sound speed propagation)
Observation::proximateFeatureofInterest:SpatialSample	The borehole trajectory
Observation::proximateFeatureofInterest:SpatialSample::sampledFeature	A part of the Earth

OMS	Geotechnical in situ test
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

B.8 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, etc.)
Observation::procedure	Instrument, analytical process (e.g. NF EN ISO 13395 October 1996/T90-012)

B.9 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 from which the core was extracted
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 was 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

Annex C (informative)

Changes in the Observation and Sample models between ISO 19156:2011 and ISO 19156:2023 (this document)

C.1 General

This annex contains information about the changes made in the Observation, Sampling and Specimen models between the previous edition of this document (ISO 19156:2011) and this second edition (ISO 19156:2023). 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.

C.2 Package and requirements class structure

The following UML packages were defined in ISO 19156:2011.

a) Observation schema

- 1) observation <>ApplicationSchema>>
- 2) measurement <>ApplicationSchema>>
- 3) categoryObservation <>RequirementsClass>>
- 4) countObservation <>RequirementsClass>>
- 5) truthObservation <>RequirementsClass>>
- 6) temporalObservation <>RequirementsClass>>
- 7) geometryObservation <>RequirementsClass>>
- 8) complexObservation <>RequirementsClass>>
- 9) coverageObservation <>RequirementsClass>>
- 10) pointCoverageObservation <>RequirementsClass>>
- 11) timeSeriesObservation <>RequirementsClass>>

b) Sampling Features

- 12) samplingFeature <>ApplicationSchema>>
- 13) spatialSamplingFeature <>ApplicationSchema>>
- 14) samplingPoint <>RequirementsClass>>

- 15) samplingCurve <<RequirementsClass>>
- 16) samplingSurface <<RequirementsClass>>
- 17) samplingSolid <<RequirementsClass>>
- 18) specimen <<RequirementsClass>>
- c) Domain specific sampling features <<informative>>
- d) Examples <<informative>>
- e) Sampling Coverage Observation <<informative>>
- f) General Feature Instance <<RequirementsClass>>
- g) Temporal Coverage <<RequirementsClass>>

The following conformance classes and the included Abstract Test Suite clauses were defined for Application Schemas in ISO 19156:2011.

- 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 19156:2023 (this document) the UML packages have been restructured to describe three levels of abstraction (conceptual schema, abstract core application schema and basic application schema) for both Observations and Samples:

- i) Conceptual Observation schema
- ii) Conceptual Sample schema
- iii) Abstract Observation Core <>ApplicationSchema>>
- iv) Abstract Sample Core <>ApplicationSchema>>
- v) Basic Observations <>ApplicationSchema>>
- vi) Basic Samples <>ApplicationSchema>>
- vii) Examples <>informative>>
- viii) Codelist realizations <>informative>>

The requirements classes of ISO 19156:2023 (this document) are much more fine-grained than in the conformance classes in ISO 19156:2011: 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 classes there is a corresponding conformance class to which a system may declare conformance. Thus, the number of conformance classes in ISO 19156:2023 (53 conformance classes) is much bigger than in ISO 19156:2011 (18 conformance classes). For the complete list of conformance classes in ISO 19156:2023, see Annex A.

C.3 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 ISO 19156:2023, although most of them capture concepts defined in ISO 19156:2011 as classes.

There are a few completely new concepts added in ISO 19156:2023:

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

- f) SampleCollection (a collection of similar Samples);
- g) ObservationCharacteristics (common characteristics of Observations);
- h) ObservingCapability (potential to create 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 ISO 19156:2011 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 ISO 19156:2011 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 ISO 19156:2011 is now captured as the Sampling interface.
- Association class PreparationStep for describing the processingDetails association role from SF_Specimen to SF_Process in ISO 19156:2011 has been remodelled as an interface PreparationStep with the processingDetails association role to the PreparationProcedure interface.

C.4 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 introduce these concepts and some related classes using the FeatureType stereotype and with more detailed sets 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.

C.5 Modelling of the Observation concept

C.5.1 OM_Observation in ISO 19156:2011

The Observation concept was modelled as OM_Observation class in ISO 19156:2011 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:

- a) featureOfInterest (Domain): GFI_Feature [1];
- b) observedProperty (Phenomenon): GF_PropertyType [1];
- c) procedure (ProcessUsed): OM_Process [1];
- d) phenomenonTime: TM_Object [1];
- e) resultTime: TM_Instant [1];
- f) result (Range): Any [1];
- g) resultQuality: DQ_Element [0..*];
- h) parameter: NamedValue [0..*];
- i) validTime: TM_Period [0..1];
- j) relatedObservation: OM_Observation [0..*];
- k) 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.

C.5.2 Observation in ISO 19156:2023

In ISO 19156:2023, the 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 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:

- i) at least one of either observer or host should be provided;
- ii) observedProperty should be a phenomenon associated with the featureOfInterest;
- iii) procedure should be suitable for the associated observedProperty;
- iv) 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:

- 1) at least one proximateFeatureOfInterest or ultimateFeatureOfInterest shall be given;
- 2) attribute and association values shall be aligned with the observationType;
- 3) exactly one observedProperty shall be given;
- 4) exactly one phenomenonTime shall be given;
- 5) exactly one observingProcedure shall be given;
- 6) exactly one result shall be given;
- 7) exactly one resultTime shall be given;
- 8) observedProperty should be a phenomenon associated with the ultimateFeatureOfInterest or the proximateFeatureOfInterest;
- 9) parameter.name shall not appear more than once;
- 10) 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 ISO 19156:2023 has the following properties with effective cardinalities and types:

- 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..*]

C.5.3 Migration from OM_Observation to Observation

An instance of the OM_Observation class of ISO 19156:2011 can be expressed as an instance of the Observation class of the Basic Observations package as follows:

- a) 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 can potentially be necessary in order to separate the ultimate and proximate features of interest.
- b) OM_Observation.observedProperty: GF_PropertyType becomes the Observation.observedProperty: ObservableProperty.
- c) 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.
- D) OM_Observation.phenomenonTime: TM_Object becomes Observation.phenomenonTime: TM_Object.
- e) OM_Observation.resultTime: TM_Instant becomes Observation.resultTime: TM_Instant.
- f) OM_Observation.result: Any becomes Observation.result: Any
- g) OM_Observation.resultQuality: DQ_Element becomes Observation.resultQuality: Any
- h) OM_Observation.parameter: NamedValue becomes Observation.parameter: NamedValue
- i) OM_Observation.validTime: TM_Period becomes Observation.validTime: TM_Period
- j) OM_Observation.relatedObservation: OM_Observation becomes Observation.relatedObservation: Observation
- k) OM_Observation.metadata: MD_Metadata [0..1] becomes Observation.metadata: Any

For information about transitioning the specialized Observation types of ISO 19156:2011 see C.8.

Table C.1 summarizes the Observation mappings from the ISO 19156:2023 Basic Observations package to ISO 19156:2011.

Table C.1 — Observation mapping from ISO 19156:2023 to ISO 19156:2011

ISO 19156:2023 class/property, Basic Observations package	Relation	ISO 19156:2011 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)

C.6 Modelling of the Sample and Sampling concepts

C.6.1 SF_SamplingFeature, SF_Specimen SF_SpatialSamplingFeature and in ISO 19156:2011

The Sampling Feature concept was modelled as SF_SamplingFeature class in ISO 19156:2011 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:

- a) sampledFeature (Intention): GFI_Feature [1..*];
- b) relatedSamplingFeature: SF_SamplingFeature [0..*], with association class SamplingFeatureComplex;
- c) relatedObservation: OM_Observation [0..*];
- d) lineage: LI_Lineage [0..1];
- e) 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:

- i) hostedProcedure (Platform): OM_Process [0..*];
- ii) positionalAccuracy: DQ_PositionalAccuracy [0..2];
- iii) 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.

C.6.2 Sample, SpatialSample, MaterialSample and StatisticalSample in ISO 19156:2023

ISO 19156:2023 introduces the Sample concept which is modelled using one interface and five classes:

- a) Sample interface in the Conceptual Sample schema package;
- b) AbstractSample class in the Abstract Sample Core package, and;
- c) Sample class and its specializations in the Basic Samples package:
 - 1) SpatialSample class;
 - 2) StatisticalSample class; and
 - 3) MaterialSample class.

The Sample interface is defined as 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:

- i) sampledFeature: Any [1..*];
- ii) relatedObservation: Conceptual Observation schema: Observation [0..*];
- iii) preparationStep: Conceptual Sample schema: PreparationStep [0..*];
- iv) sampling: Conceptual Sample schema: Sampling [0..*];
- v) relatedSample: Conceptual Sample schema: Sample [0..*];
- vi) sampleType: AbstractSampleType [0..*];
- vii) parameter: NamedValue [0..*];
- viii) 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:

- 1) SpatialSample adds the following attributes:
 - shape: Geometry [0..1];
 - horizontalPositionalAccuracy: Any [0..1];
 - verticalPositionalAccuracy: Any [0..1].
- 2) StatisticalSample adds the following attribute:
 - classification: StatisticalClassification [0..*].
- 3) MaterialSample adds the following attributes:
 - size: PhysicalDimension [0..*];
 - sourceLocation: Geometry [0..1];
 - storageLocation: NamedLocation [0..1].

C.6.3 Modelling of environmental monitoring stations

In ISO 19156:2011, the SF_SamplingPoint 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 ISO 19156:2023 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.observingProcedure: ObservingProcedure association.

C.6.4 Migration from SF_SamplingFeature to Sample

An instance of SF_SamplingFeature class of ISO 19156:2011 can be expressed as an instance of the Sample class of the Basic Samples package as follows:

- a) SF_SamplingFeature.sampledFeature: GFI_Feature becomes Sample.sampledFeature: Any;
- b) SF_SamplingFeature.relatedSamplingFeature: SF_SamplingFeature becomes Sample.relatedSample: Conceptual Sample schema: Sample; the value role:GenericName attribute of association class

SamplingFeatureComplex becomes the value of the context:GenericName qualifier of the relatedSample association;

- c) SF_SamplingFeature.relatedObservation: OM_Observation becomes Sample.relatedObservation: Conceptual Sample schema: Observation;
- d) SF_SamplingFeature.lineage: LI_Lineage is expressed with Sample.metadata: Any;
- e) SF_SamplingFeature.parameter: NamedValue becomes Sample.parameter: NamedValue.

Table C.2 summarizes the Sample mappings from the ISO 19156:2023 Basic Samples package to ISO 19156:2011.

Table C.2 — Sample mapping from ISO 19156:2023 to ISO 19156:2011

ISO 19156:2023 class/property, Basic Samples package	Relation	ISO 19156:2011 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)

C.6.5 Migration from SF_SpatialSamplingFeature to SpatialSample

An instance of SF_SpatialSamplingFeature class of ISO 19156:2011 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):

- a) SF_SpatialSamplingFeature.hostedProcedure: OM_Process becomes the Observer.observingProcedure: ObservingProcedure (observing procedures no longer associated with sampling features);
- b) 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 ISO 19156:2011 see C.8.

Table C.3 summarizes the SpatialSample mappings from the ISO 19156:2023 Basic Samples package to ISO 19156:2011, including the properties inherited from the Sample and SF_SamplingFeature.

Table C.3 — SpatialSample mapping from ISO 19156:2022 to ISO 19156:2011

ISO 19156:2023 class/property, Basic Samples package	Relation	ISO 19156:2011 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

C.6.6 Migration from SF_Specimen to MaterialSample

An instance of SF_Specimen class of ISO 19156:2011 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):

- a) 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:
 - 1) 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;
 - 2) PreparationStep.time: TM_Object becomes AbstractPreparationStep.time: TM_Object.
- b) SF_Specimen.currentLocation: Location becomes MaterialSample.storageLocation: NamedLocation;
- c) SF_Specimen.materialClass: GenericName is expressed using the AbstractSample.sampleType with appropriate code list values for sample material classification;
- d) SF_Specimen.samplingLocation: GM_Object becomes MaterialSample.sourceLocation: Geometry and/or Sampling.samplingLocation: Geometry via the MaterialSample.sampling association;
- e) SF_Specimen.samplingMethod: SF_Process becomes the Sampling.samplingProcedure: Conceptual Sample schema: SamplingProcedure via the MaterialSample.sampling association;
- f) SF_Specimen.samplingTime: TM_Object becomes Sampling.time: TM_Object via the MaterialSample.sampling association;
- g) SF_Specimen.size: Measure becomes MaterialSample.size: PhysicalDimension (multiple named size qualifiers possible with a dimension and value: Measure for each);
- h) SF_Specimen.specimenType: GenericName becomes AbstractSample.sampleType with appropriate code list values for domain specific sample classification (several sample types allowed).

Table C.4 summarizes the MaterialSample mappings from the ISO 19156:2023 Basic Samples package to ISO 19156:2011, including the properties inherited from the Sample and SF_SamplingFeature.

Table C.4 — MaterialSample mapping from ISO 19156:2023 to ISO 19156:2011

ISO 19156:2023 class/property, Basic Samples package	Relation	ISO 19156:2011 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

C.7 Observation and Sample collections

ISO 19156:2011 did not include a concept of an Observation collection. In ISO 19156:2023 it is added as class AbstractObservationCollection in package Abstract Observation Core with the following attributes, associations and cardinalities:

- a) member: Conceptual Observation schema: Observation [0..*];
- b) memberCharacteristics: AbstractObservationCharacteristics [0..1];
- c) collectionType: AbstractObservationCollectionType [0..*];
- d) relatedCollection: AbstractObservationCollection [0..*];
- e) metadata: Any [0..*].

A concrete specialization of the AbstractObservationCollection class is provided in the Basic Observations package as the ObservationCollection class. The same package also contains one concrete specialization of the AbstractObservationCollectionType codelist, the ObservationCollectionType class, which has an initial set of two values: 'homogeneous' and 'summarizing' defining how the properties of the ObservationCharacteristics instances associated with the ObservationCollection instance relate to the corresponding properties of the collection members (see 10.12.1). Other ObservationCollection classifications may be added by specializing the AbstractObservationCollectionType as required.

ISO 19156:2011 provided a collection of Sampling features as SF_SamplingFeatureCollection class with a single association role member: SF_SamplingFeature. In ISO 19156:2023, 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.

Table C.5 summarizes the SampleCollection mappings from the ISO 19156:2023 Basic Samples package to ISO 19156:2011.

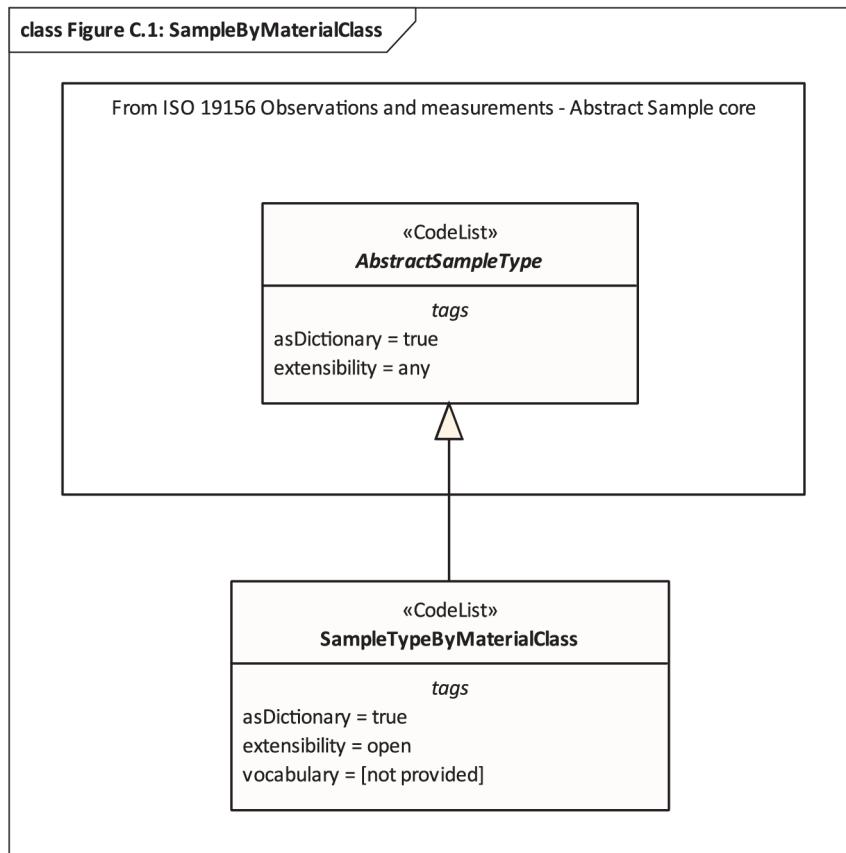
Table C.5 — SampleCollection mapping from ISO 19156:2023 to ISO 19156:2011

ISO 19156:2023 class/property, Basic Samples package	Relation	ISO 19156:2011 class/property
SampleCollection	equivalent class	SF_SamplingFeatureCollection
SampleCollection.member	equivalent property	SF_samplingFeatureCollection.member
SampleCollection.relatedCollection		(no match)

C.8 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 19156:2011 are modelled using soft-typing-based classification schemes in ISO 19156:2023 (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.

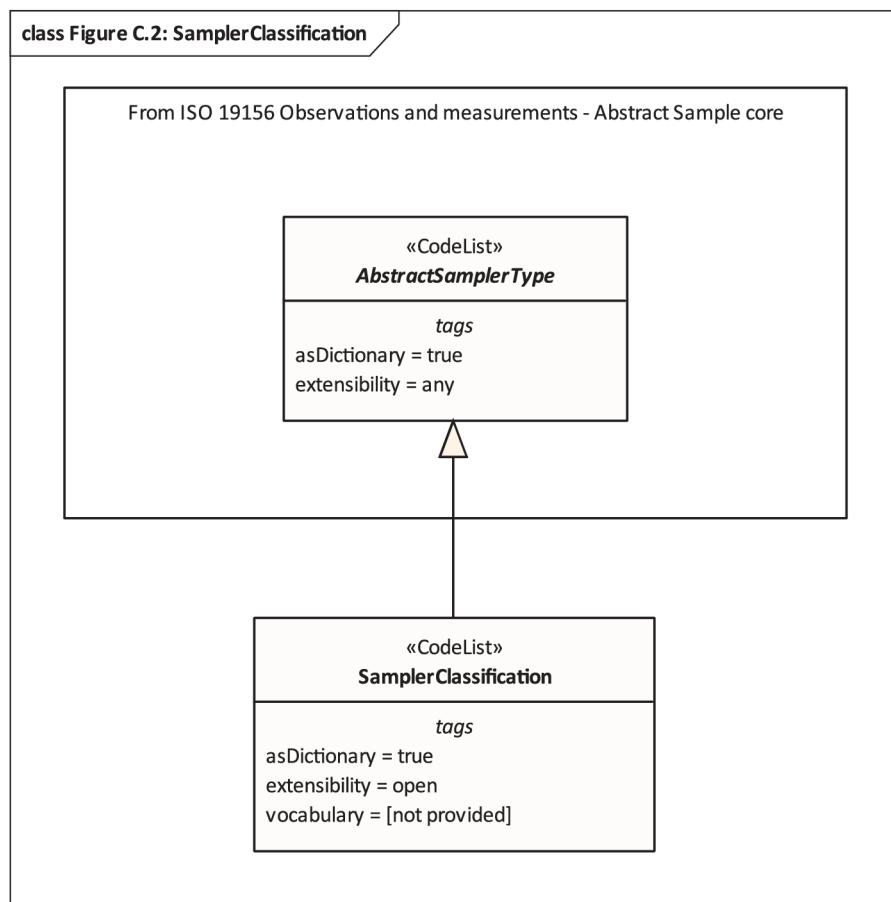
Concrete codelists are provided for both the result-type based Observation classification and the shape geometry SpatialSample classification. Any additional Observation and Sample classification schemes can be provided in the domain models by extending the AbstractObservationType and AbstractSampleType classes, as illustrated for classification of Samples in Figure C.1.



NOTE No values or vocabulary are provided for SampleTypeByMaterialClass in this document. Classes are provided here only as an example of the codelist extension mechanism for application-domain-specific implementations.

Figure C.1 — (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 C.2.



NOTE No values or vocabulary are provided for SamplerClassification in this document. Classes are provided here only as an example of the codelist extension mechanism for application-domain-specific implementations.

Figure C.2 — (Example) Mechanism for defining a generic classification scheme for Samplers by extending the AbstractSamplerType codelist

C.8.1 Migration of result-type-based Observation types

Instances of the specialized Observation types of ISO 19156:2011 can be migrated into instances of the ISO 19156:2023 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):

- a) OM_Observation: Observation;
- b) OM_Measurement: Measurement;
- c) OM_CategoryObservation: Category Observation;
- d) OM_CountObservation: Count Observation;
- e) OM_TruthObservation: Truth Observation;
- f) OM_TemporalObservation: Temporal Observation;
- g) OM_GeometryObservation: Geometry Observation;
- h) OM_ComplexObservation: Complex Observation;
- i) OM_DiscreteCoverageObservation: Discrete CoverageObservation;
- j) OM_PointCoverageObservation: Point Coverage Observation;
- k) OM_TimeSeriesObservation: Time Series Observation.

C.8.2 Migration of geometry-based sampling feature types

Instances of the specialized sampling feature types of ISO 19156:2011 can be migrated into instances of the ISO 19156:2023 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):

- a) SF_SamplingPoint: Point Sample;
- b) SF_SamplingCurve: Curve Sample;
- c) SF_SamplingSurface: Surface Sample;
- d) SF_SamplingSolid: Solid Sample.

C.9 Generic metadata associations

In ISO 19156:2011 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. ISO 19156:2023 allows for providing metadata in addition to the concepts covered by the OMS model for most of the model classes:

- a) Abstract Observation Core package:
 - 1) AbstractObservationCharacteristics;
 - 2) AbstractObservationCollection;
 - 3) AbstractObservingProcedure;

- 4) AbstractObservableProperty;
 - 5) AbstractObserver;
 - 6) AbstractDeployment;
 - 7) AbstractHost.
- b) Abstract Sample Core package:
- 1) AbstractSample;
 - 2) AbstractSampling;
 - 3) AbstractSampler;
 - 4) AbstractPreparationStep;
 - 5) AbstractPreparationProcedure;
 - 6) AbstractSamplingProcedure.
- c) Basic Samples
- 1) SampleCollection.

Each of these classes contains an attribute with role name metadata of type Any and with cardinality of 0..*. ISO 19115 series metadata records may still be used for providing Observation instance metadata, but it is no longer the only allowed metadata model. With this change, ISO 19115² is also no longer a normative reference of the ISO 19156:2023.

C.10 Discarded concepts

The ISO 19156:2011 contained two requirementsClass packages with classes used in the UML but not specific to the Observations and Sampling features:

- a) General Feature Instance package:
 - 1) GFI_DomainFeature;
 - 2) GFI_Feature.
- b) Temporal Coverage package:
 - 1) CVT_DiscreteTimeInstantCoverage;
 - 2) CVT_TimeInstantValuePair.

The General Feature Instance package and its contained classes are not included in ISO 19156:2023, as the General feature instances are no longer required in either the Observation or Sample models.

² Withdrawn. Replaced by ISO 19115-1:2014.

The Temporal Coverage package and its contained classes are not included in ISO 19156:2023, 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 ISO 19156:2011 UML model will be revised to profile the ISO 19156:2023 model instead, and to provide a detailed conceptual model for Observations with temporal coverage type results.

Annex D

(informative)

Best practices in use of the Observation and Sampling models

D.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 the ISO 19101-1 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 can 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 19123-1. This viewpoint focuses on the variation of a property within the (spatiotemporal) domain of interest. The domain can 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 can 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 can 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 D.1, 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 can be the result of an Observation.

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

- a) 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.
- b) 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.
- c) 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 for which it provides property information, the Feature representation can 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.

Location	Properties			
	Property 1	Property 2	...	Property m
(x ₁ , y ₁)	Value ¹ ₁	Value ² ₁	...	Value ^m ₁
(x ₂ , y ₂)	Value ¹ ₂	Value ² ₂	...	Value ^m ₂
Feature 3	(x ₃ , y ₃)	Value ¹ ₃	Value ² ₃	...

	(x _n , y _n)	Value ¹ _n	Value ² _n	...
				Value ^m _n
	Coverage 2			

Figure D.1 — Tabular representation of information associated with a set of locations

D.2 Observation concerns

D.2.1 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 can 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 can 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 also need to 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:

- a) a property of each distinct material sample on which atomic observations are actually made;
- b) a property of the sampling site (which would require distinct sites for all elevations at which observations are made);
- c) a parameter of the observation procedure (which makes the procedure specific to this observation series only), or;
- d) 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" can be derived from an Observation, fixing the observedProperty to be "ChemicalConcentration" and adding an additional attribute "analyte".

D.2.2 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^[20] 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 can 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 can 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 can 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.

D.2.3 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 can 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, whether from the previous generation (OGC SOS^[21]) or the current one (OGC SensorThings API^[22]), share commonalities in the way they approach this topic. They both leverage the OMS model to directly allow filtering on featureOfInterest, observedProperty and procedure.

- a) 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).
- b) SOS,^[21] 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 can 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 can appear inline, perhaps described using one of the models presented here, or they can be indicated by reference using a URI.^[13] The URI identifier can be a URL link or service call, which should resolve immediately to yield a complete resource. Or, it can 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.

D.2.4 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 degrees. 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 (e.g. 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" and "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 document, 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.

D.3 Sample, Sampling concerns

D.3.1 Sample as observation-collector

The Sample model provides:

- a) an intermediate Sample class that allows the assignment of primitive and intermediate properties within a processing chain;
- b) three sub-types of Samples corresponding to practices applied by communities where Samples are either defined by their geospatial characteristics, statistical characteristics or their material ones (being taken ex-situ for further observation); and
- c) additional classes providing a context for the description of sampling acts and regimes.

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 can organize its data according to flightlines, ship's tracks, outcrops, sampling-stations, quadrats, etc. or an observation archive or museum can organize observations by specimen (a specific type of material sample targeting preservation).

D.3.2 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 is not always 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 is required to 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 necessarily has 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 can help further refine the understanding of the type of the ultimate feature-of-interest.

EXAMPLE 1 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.

EXAMPLE 2 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.

EXAMPLE 3 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 analyses are conducted in a laboratory.

D.3.3 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 7.2.2.2 the relationship between the observed property and property(ies) of the ultimate feature-of-interest is often more complex.

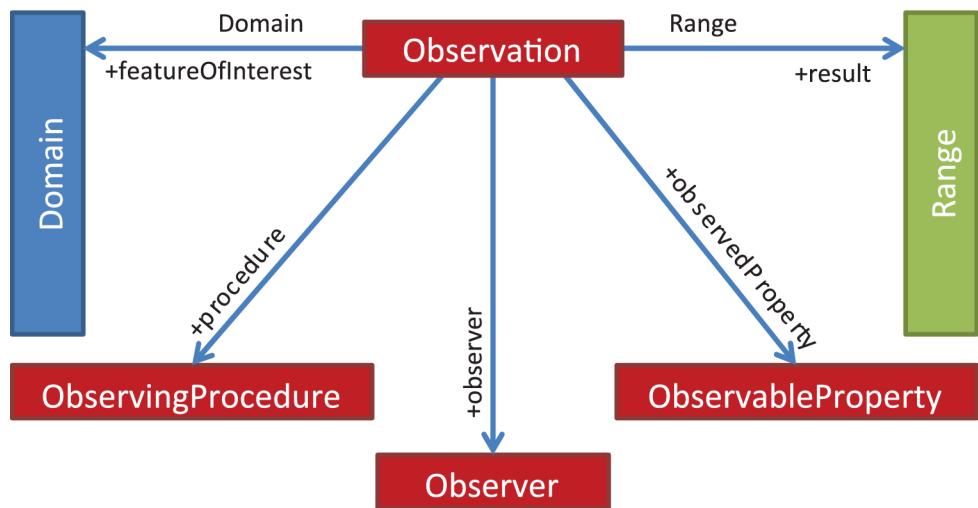
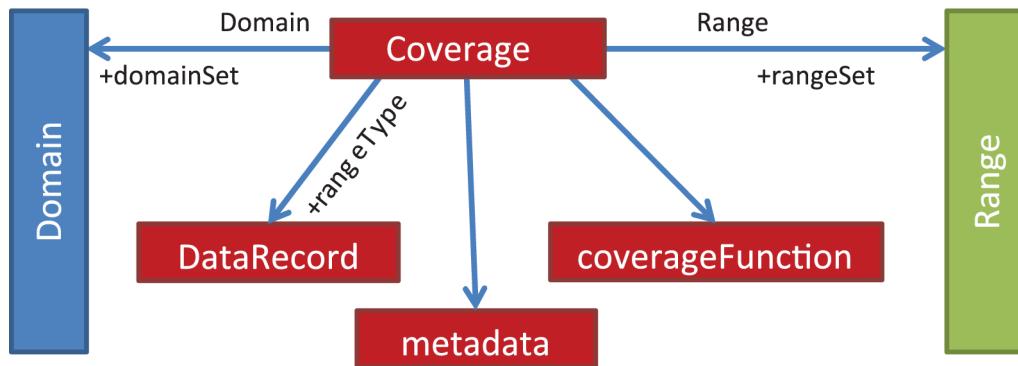
The Sample model is a mechanism for preserving the strict association, by providing a specific intermediate feature type whose observable properties are unspecified in advance, but supplied through an unlimited set of related Observations. The path from a sensed property obtained through Observations 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 is required to apply to a feature-of-interest that bears this property, or a sampling feature. In some cases, the types of these features are not conventional or immediately recognizable, but the coherence of the OMS model does imply their existence. Hence, if any intermediate result is made explicit, then a suitable intermediate feature also needs to be identified.

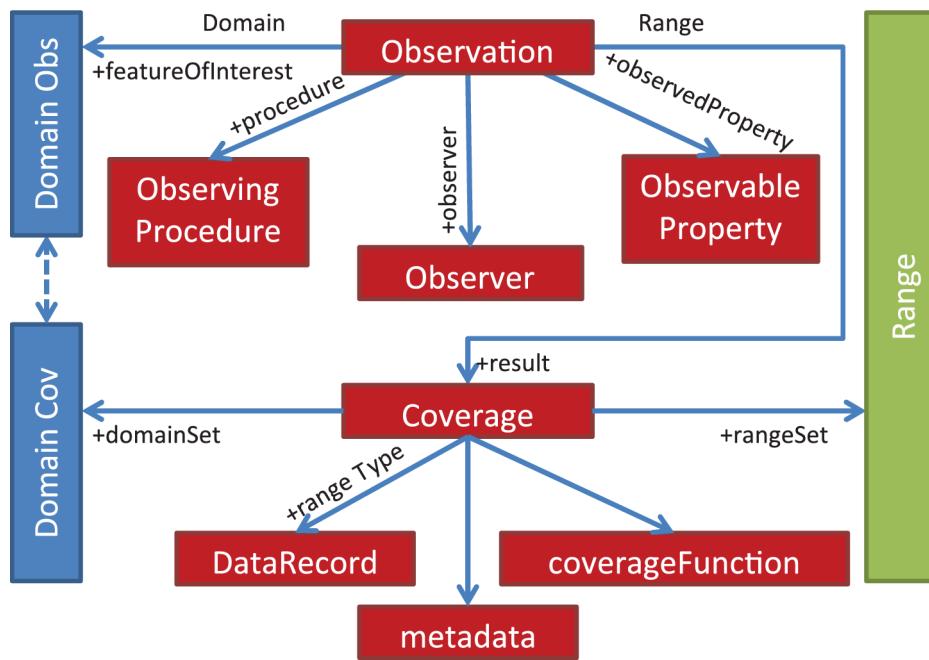
D.4 Observations and Coverages

Within the Open Geospatial Consortium (OGC), different data models have evolved for the provision of sensor data (OMS model) and datacubes [OGC Coverage Implementation Schema (ISO 19123-2 and ISO 19123-1) (CIS)]. While these models are formally distinct, and were developed mostly independently of each other, there are great similarities as well as overlaps between them. 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.

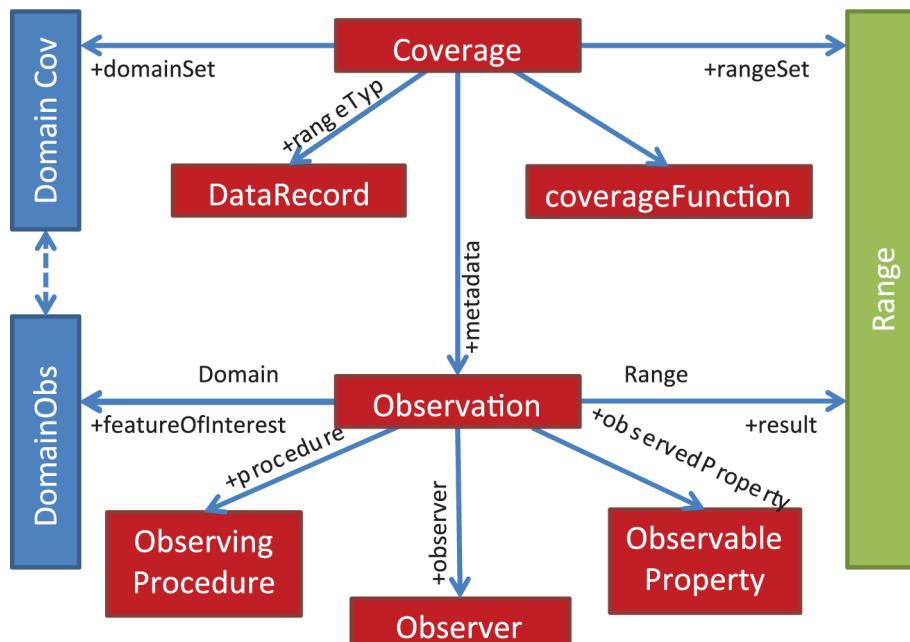
Upon closer analysis it becomes clear just how similar these models are at their core, as well as how each provides concepts essential for the precise description of the data that have been neglected within the other models. Both OMS and CIS provide a set of values (Range) over a given extent (Domain). However, while the CIS model (Figure D.3) 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 (Figure D.2) provides far more detailed information on the measurement methodology and process via the ObservableProperty, ObservingProcedure and Observer types.

**Figure D.2 — OMS model key elements****Figure D.3 — CIS model key elements**

When OMS and CIS models are used in conjunction, care needs to 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) can reference a feature representing this transect or profile (Figure D.4), 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 D.4 — Coverage as a result of an Observation**

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 (Figure D.5), whereby the Observation result was left as void. In this document, 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 be an envelope of the CIS Domain.

**Figure D.5 — Observation as metadata of a coverage**

Annex E (informative)

Detailed package overview diagrams

E.1 General

The UML class diagrams in this Annex are provided for additional reference in cases where a complete picture of all classes contained in a package is useful. They are provided here despite the fact that the text is most likely not readable with typical A4 format print resolution. The intended use is for online browsing with zoom-in capability or for printing in high-resolution.

E.2 Abstract Observation Core — Overview

The Figure E.1 provides a diagram of all classes in package Abstract Observation Core. This figure is also made available as a standalone PDF document at <https://standards.iso.org/iso/19156/ed-2/en>.

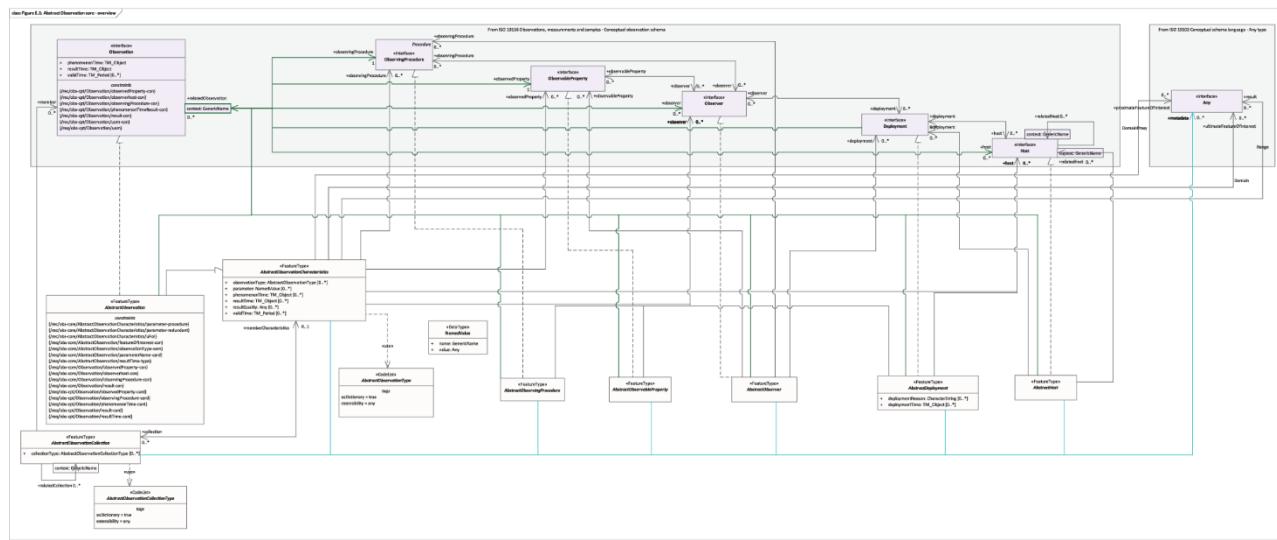


Figure E.1 — Abstract Observation Core – Overview

E.3 Basic Observations — Overview

The Figure E.2 provides a diagram of all classes in package Basic Observations. This Figure is also made available as a standalone PDF document at <https://standards.iso.org/iso/19156/ed-2/en>.

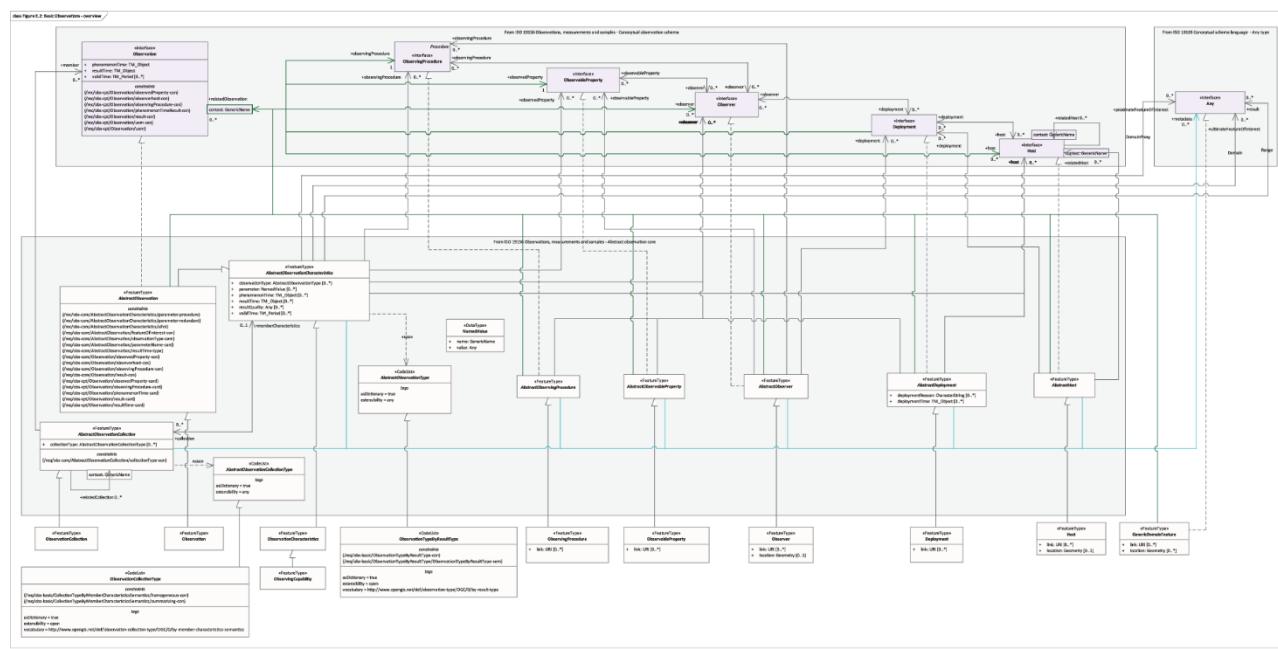


Figure E.2 — Basic Observations – Overview

E.4 Abstract Sample Core — Overview

The Figure E.3 provides a diagram of all classes in package Abstract Sample Core. This figure is also made available as a standalone PDF document at <https://standards.iso.org/iso/19156/ed-2/en>.

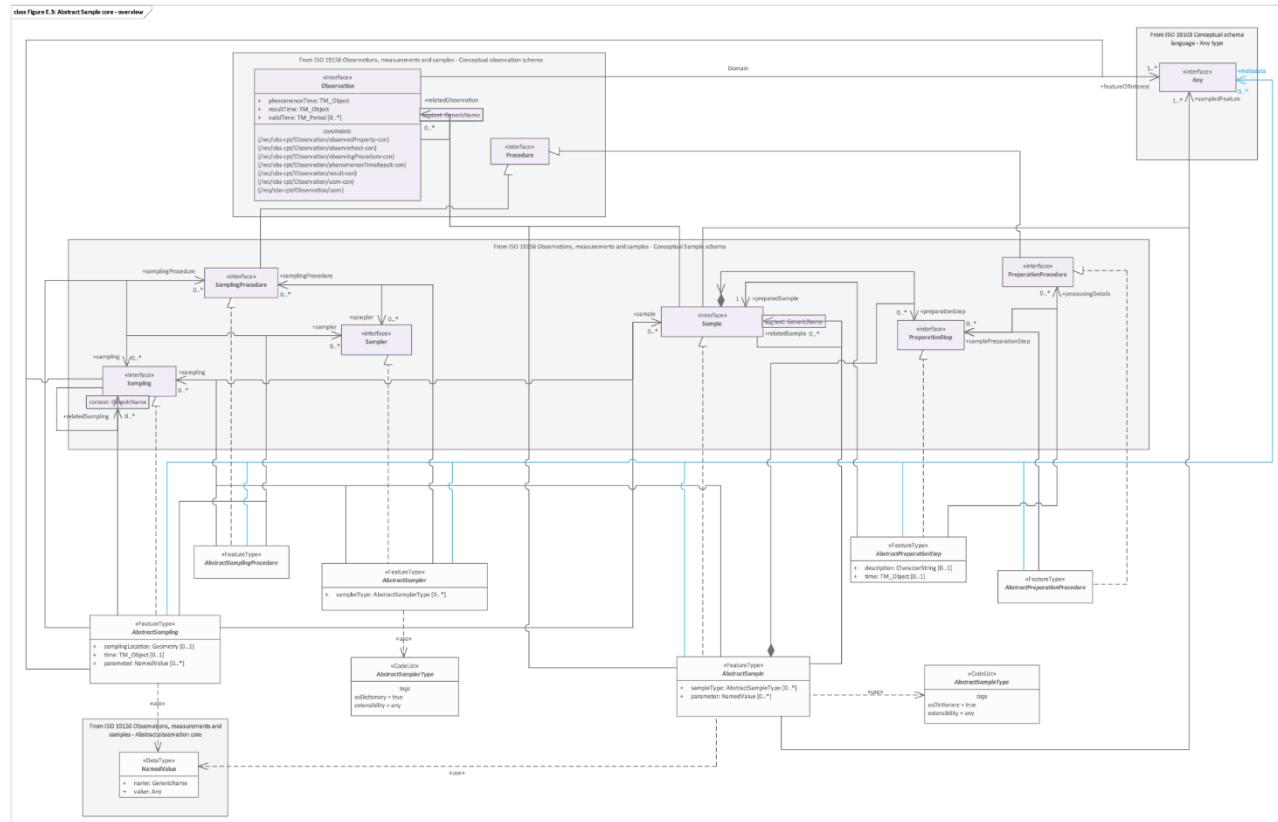


Figure E.3 — Abstract Sample Core – Overview

E.5 Basic Samples — Overview

Figure E.4 provides a diagram of all classes in package Abstract Sample Core. This figure is also made available as a standalone PDF document at <https://standards.iso.org/iso/19156/ed-2/en>.

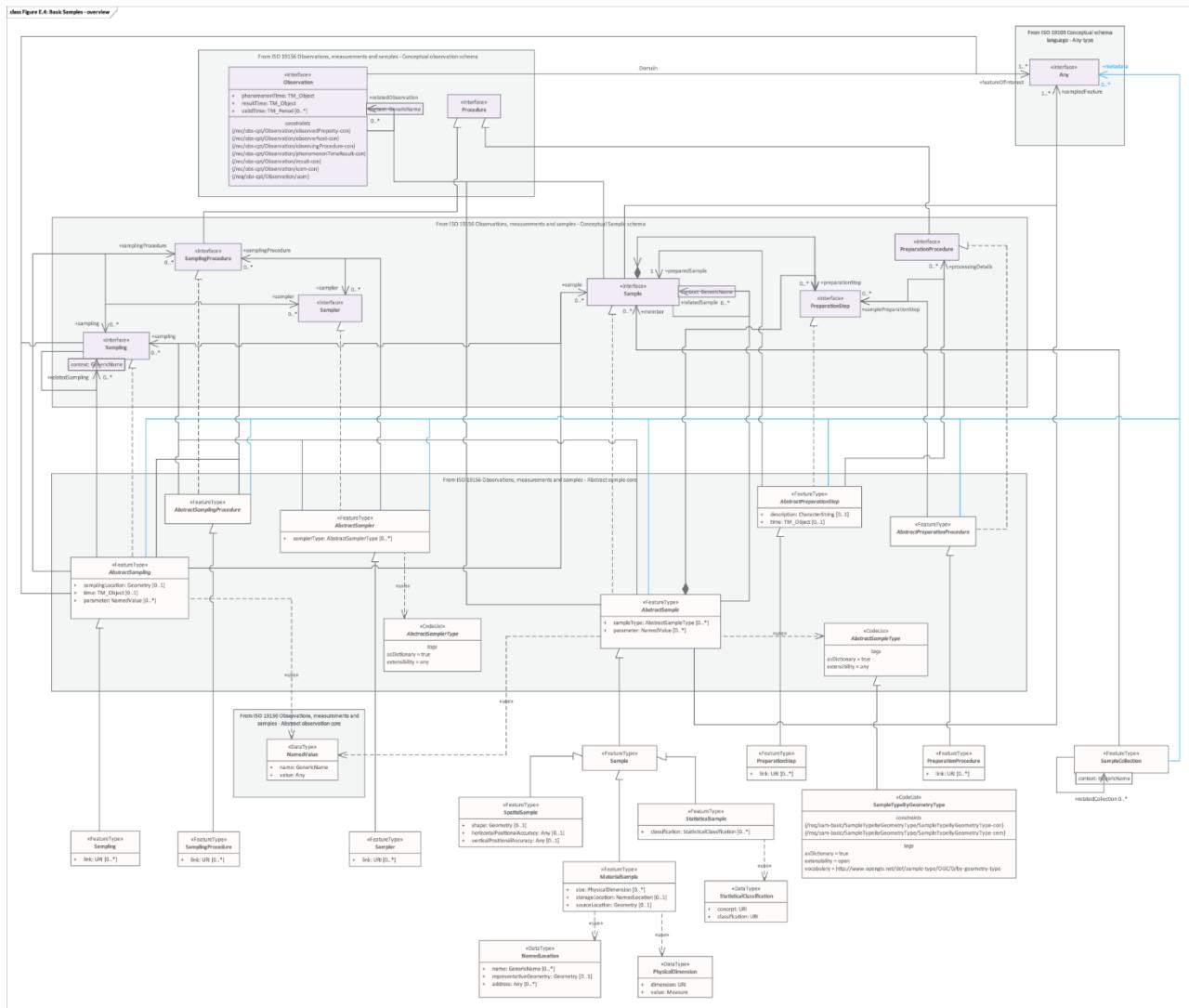


Figure E.4 — Basic Samples – Overview

Bibliography

- [1] ISO 19101-1:2014, *Geographic information — Reference model — Part 1: Fundamentals*
- [2] ISO 19109:2015, *Geographic information — Rules for application schema*
- [3] ISO 19115-1, *Geographic information — Metadata — Part 1: Fundamentals*
- [4] ISO 19115-1:2014/Amd 2:2020, *Geographic information — Metadata — Part 1: Fundamentals — Amendment 2*
- [5] ISO 19123-1:³, *Geographic information — Schema for coverage geometry and functions — Part 1 Fundamentals*
- [6] ISO 19123-2, *Geographic information — Schema for coverage geometry and functions — Part 2: Coverage implementation schema*
- [7] ISO 19136-1:2020, *Geographic information — Geography Markup Language (GML) — Part 1: Fundamentals*
- [8] ISO 19157, *Geographic information — Data quality*
- [9] ISO 19157:2013/Amd 1:2018, *Geographic information — Data quality — Amendment 1: Describing data quality using coverages*
- [10] ISO 19157-1, *Geographic information — Data quality — Part 1: General requirements*
- [11] CHRISMAN N.R. *Exploring Geographical Information Systems*. Wiley, Second Edition, 2001
- [12] FOWLER M. *Analysis Patterns: reusable object models*. Addison Wesley Longman, Menlo Park, CA, 1998
- [13] IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*. August 1998
- [14] JCGM 200:2012, *International vocabulary of metrology — Basic and general concepts and associated terms (VIM)*
- [15] KRANTZ D.H., LUCE R.D., SUPPES P., TVERSKY A. *Additive and polynomial representations*. Foundations of measurement. Academic Press, New York, Vol. I, 1971
- [16] LUCE R.D., KRANTZ D.H., SUPPES P., TVERSKY A. *Representation, axiomatization, and invariance*. Foundations of measurement. Academic Press, New York, Vol. III, 1990
- [17] 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>

³ Under preparation. Stage at the time of publication: ISO/FDIS 19123-1:2023.

- [18] 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
- [19] 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)
- [20] *Sensor Model Language (SensorML)*, OpenGIS® Implementation Standard, OGC 12-000r2. Available (viewed 2020-09-29) at <http://www.opengeospatial.org/standards/sensorml>
- [21] *Sensor Observation Service*, OpenGIS® Implementation Specification OGC document 12-006
- [22] The OGC SensorThings API Part 1: Sensing (2016). OGC Document OGC: 15-078R6,
- [23] SUPPES P., KRANTZ D.H., LUCE R.D., TVERSKY A. *Geometrical, threshold, and probabilistic representations*. Foundations of measurement. Academic Press, New York, Vol. II, 1989
- [24] *SWE Common Data Model Encoding Standard*, OpenGIS® Implementation Standard OGC document 08094r1
- [25] OGC: The Specification Model - A Standard for Modular specifications (2009). OGC document 08-131r3,
- [26] SCHLEIDT K., BAUMANN P. "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.
- [27] *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/>
- [28] *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/>
- [29] *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>
- [30] *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>
- [31] SPATIAL DATA ON THE WEB BEST PRACTICES. W3C Working Group Note, 28 September 2017. Also published as OGC Best Practice 15-107, <https://www.w3.org/TR/sdw-bp/>
- [32] ISO 19116:2019, *Geographic information — Positioning services*
- [33] ISO/IEC 19501:2005, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

- [34] ISO 19143:2010, *Geographic information — Filter encoding*
- [35] ISO 9000:2015, *Quality management systems — Fundamentals and vocabulary*
- [36] ISO 19101-2:2018, *Geographic information — Reference model — Part 2: Imagery*