Open Geospatial Consortium Inc.

Date: 2010-09-02

Reference number of this document: OGC 10-037

OGC name of this OGC® project document: http://www.opengis.net/doc/IS/SOS/2.0

Version: 2.0.0

Category: OpenGIS® Interface Standard

Editor: Arne Broering (52°North)

Christoph Stasch (IfGI)

Johannes Echterhoff (iGSI)

OGC® SOS 2.0 Interface Standard

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

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

Document type: OpenGIS® Interface Standard

Document subtype:

Document stage: Draft proposed version

Document language: English

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

Contents Page

1 Scope 1

2 Compliance 1

3 Normative references 3

4 Terms and definitions 4

4.1 Measurement 4

4.2 Observable 4

4.3 Observation Offering 4

4.4 Procedure 4

4.5 Result 4

5 Conventions 5

5.1 Abbreviated terms 5

5.2 UML notation 5

5.3 Platform-neutral and platform-specific standards 5

5.4 Data dictionary tables 6

5.5 Classes imported from other specifications with predefined XML encoding 6

5.6 Namespace Conventions 7

6 Observation Model Overview 8

7 SOS Model Overview 8

7.1 Workflow of Observation Retrieval 10

7.2 Workflow of Observation Insertion 11

7.3 Workflow of Result Insertion 12

7.4 Workflow of Result Retrieval 13

8 Requirements Class: Core 14

8.1 GetCapabilities Operation 15

8.1.1 Request 16

8.1.2 Response 17

8.1.2.1 FilterCapabilities Section 19

8.1.2.2 Contents Section 20

8.1.3 Exceptions 26

8.1.4 Examples 26

8.2 DescribeSensor Operation 26

8.2.1 Examples 27

8.3 GetObservation Operation 27

8.3.1 Request 28

8.3.2 Response 30

8.3.3 Exceptions 31

8.3.4 Examples 31

8.4 Codes 32

8.4.1 ValueCode 33

9 Enhanced Operations Extension 34

9.1 Requirements Class: Feature of Interest Retrieval 34

9.1.1 GetFeatureOfInterest Operation 34

9.1.1.1 Request 34

9.1.1.2 Response 36

9.1.1.3 Exceptions 36

9.1.1.4 Examples 37

9.2 Requirements Class: Observation Retrieval By ID 37

9.2.1 GetObservationById operation 37

9.2.1.1 Request 38

9.2.1.2 Response 38

9.2.1.3 Exceptions 39

9.2.1.4 Examples 40

10 Transactional Extension 40

10.1 Requirements Class: InsertionCapabilities 40

10.1.1 InsertionCapabilities Section 40

10.2 Requirements Class: Sensor Insertion 42

10.2.1 InsertSensor Operation 42

10.2.1.1 Request 43

10.2.1.2 Response 44

10.2.1.3 Exceptions 44

10.2.1.4 Examples 45

10.3 Requirements Class: Sensor Deletion 45

10.3.1 DeleteSensor Operation 45

10.3.1.1 Exceptions 46

10.3.1.2 Examples 46

10.4 Requirements Class: Observation Insertion 46

10.4.1 InsertObservation Operation 47

10.4.1.1 Request 47

10.4.1.2 Response 48

10.4.1.3 Exceptions 49

10.4.1.4 Examples 50

11 Result Handling Extension 50

11.1 Requirements Class: Result Insertion 50

11.1.1 InsertResultTemplate Operation 50

11.1.1.1 Request 51

11.1.1.2 Response 54

11.1.1.3 Exceptions 55

11.1.1.4 Examples 55

11.1.2 InsertResult Operation 56

11.1.2.1 Request 56

11.1.2.2 Response 57

11.1.2.3 Exceptions 58

11.1.2.4 Examples 58

11.2 Requirements Class: Result Retrieval 59

11.2.1 GetResultTemplate Operation 59

11.2.1.1 Request 60

11.2.1.2 Response 61

11.2.1.3 Exceptions 62

11.2.1.4 Examples 62

11.2.2 GetResult Operation 62

11.2.2.1 Request 63

11.2.2.2 Response 65

11.2.2.3 Exceptions 65

11.2.2.4 Examples 66

12 Spatial Filtering Profile 66

12.1 Requirements Class: Spatial Filtering Profile 66

13 Binding Extension 68

13.1 Requirements Class: XML Encoding 68

13.2 Requirements Class: Core KVP Binding 71

13.2.1 GetCapabilities KVP Binding 72

13.2.2 DescribeSensor KVP Binding 72

13.2.3 GetObservation KVP Binding 74

13.3 Requirements Class: GetFeatureOfInterest KVP Binding 76

13.4 Requirements Class: SOAP Binding 77

13.4.1 Exceptions 78

13.4.2 Action URIs 78

13.4.3 SOAP Message Body 80

13.4.4 Example 80

14 Annex A – Abstract test suite (normative) 82

14.1 SOS Core Tests 82

14.1.1 Conformance Class: SOS Core 82

14.2 SOS Enhanced Operations Extension Tests 93

14.2.1 Conformance Class: SOS Feature of Interest Retrieval 93

14.2.2 Conformance Class: SOS Observation Retrieval By ID 95

14.3 SOS Transactional Extension Tests 96

14.3.1 Conformance Class: SOS Insertion Capabilities 97

14.3.2 Conformance class: SOS Sensor Insertion 97

14.3.3 Conformance Class: SOS Sensor Deletion 99

14.3.4 Conformance Class: SOS Observation Insertion 101

14.4 SOS Result Handling Extension 104

14.4.1 Conformance Class: Result Insertion 104

14.4.2 Conformance Class: Result Retrieval 111

14.5 SOS Spatial Filtering Profile 114

14.5.1 Conformance Class: Spatial Filtering Profile 114

14.6 SOS Binding Extension Tests 115

14.6.1 Conformance Class: XML Encoding 115

14.6.2 Conformance Class: KVP Binding Extension 116

14.6.3 Conformance Class: GetFeatureOfInterest KVP Binding 119

14.6.4 Conformance Class: SOAP Binding 120

15 Annex B - Phenomena and Units of Measure (informative) 122

B.1 Identifying and referencing Phenomena and Units of Measure 122

B.2 Describing and defining Phenomena and Units of Measure 122

16 Annex C - Relationship to Other OGC Web Service Standards (informative) 124

C.1 Relationship to Web Feature Service 124

**Figures** Page

[Figure 1: O&M Basic Observation Model 8](#_Toc271208939)

[Figure 2: SOS core, extensions and OGC/SWE specification basis 9](#_Toc271208940)

[Figure 3: Operation sequence for observation retrieval 11](#_Toc271208941)

[Figure 4: Operation sequence for observation insertion 12](#_Toc271208942)

[Figure 5: Sequence of operations for insertion of results 13](#_Toc271208943)

[Figure 6: Sequence diagram of result retrieval 13](#_Toc271208944)

[Figure 7: Data types of the GetCapabilities operation 16](#_Toc271208945)

[Figure 8: Data types of Contents section 22](#_Toc271208946)

[Figure 9: Data types of GetObservation operation 28](#_Toc271208947)

[Figure 10: Data types of Codes package 32](#_Toc271208948)

[Figure 11: Data types of GetFeatureOfInterest operation 34](#_Toc271208949)

[Figure 12: Data types of GetObservationById operation 38](#_Toc271208950)

[Figure 13: Data type of InsertionCapabilities section 41](#_Toc271208951)

[Figure 14: Data types of InsertSensor operation 43](#_Toc271208952)

[Figure 15: Data types of InsertObservation operation 47](#_Toc271208953)

[Figure 16: Data types of InsertResultTemplate operation 51](#_Toc271208954)

[Figure 17: Data types of InsertResult operation 56](#_Toc271208955)

[Figure 18: Data types of GetResultTemplate operation 60](#_Toc271208956)

[Figure 19: Data types of GetResult operation 63](#_Toc271208957)

[Figure 16‑1. Relationship between SOS and WFS where WFS is providing features of interest 124](#_Toc271208958)

[Figure 16‑2. Relationship between SOS and WFS where SOS is encapsulating WFS 125](#_Toc271208959)

**Tables** Page

[Table 1: Main Requirements classes 1](#_Toc271208960)

[Table 2: Requirements classes in Transactional Extension 2](#_Toc271208961)

[Table 3: Requirements classes in Result Handling in Extension 2](#_Toc271208962)

[Table 4: Requirements classes in Enhanced Operations Extension 2](#_Toc271208963)

[Table 5: Requirements classes in Spatial Filtering Profile 2](#_Toc271208964)

[Table 6: Requirements classes in Binding Extension 3](#_Toc271208965)

[Table 7: XML Schema implementation of types from the SWE Service Model [OGC 09-001] 6](#_Toc271208966)

[Table 8: Implementation of types from OWS Common [OGC 06-121r3] 6](#_Toc271208967)

[Table 9: Implementation of types from SWE Common Data Model [OGC 08-094] 7](#_Toc271208968)

[Table 10: Implementation of types from ISO 19143 (Filter Encoding Specification) 7](#_Toc271208969)

[Table 11: Prefixes and Namespaces used in this standard 7](#_Toc271208970)

[Table 12: Properties in the GetCapabilities data type 17](#_Toc271208971)

[Table 13: Additional section names for SOS Capabilities 17](#_Toc271208972)

[Table 14: Properties in the Capabilities data type 18](#_Toc271208973)

[Table 15: Properties of Contents data type 23](#_Toc271208974)

[Table 16: Properties of ObservationOffering data type 24](#_Toc271208975)

[Table 17: Inheritance of ObservationOffering properties (from Contents) 25](#_Toc271208976)

[Table 18: Properties of GetObservation request data type 29](#_Toc271208977)

[Table 19: Properties of GetObservationResponse data type 31](#_Toc271208978)

[Table 20 – List of some code values used for identifying types defined in a conceptual model 33](#_Toc271208979)

[Table 21: Properties of GetFeatureOfInterest data type 36](#_Toc271208980)

[Table 22: Properties of GetFeatureOfInterestResponse data type 36](#_Toc271208981)

[Table 23: Properties of GetObservationById data type 38](#_Toc271208982)

[Table 24: Properties of GetObservationByIdResponse data type 39](#_Toc271208983)

[Table 25: Properties of InsertionCapabilities data type 41](#_Toc271208984)

[Table 26 Properties of SosInsertionMetadata data type 44](#_Toc271208985)

[Table 27: Properties of InsertObservation data type 48](#_Toc271208986)

[Table 28: Properties of InsertObservationResponse data type 49](#_Toc271208987)

[Table 29: Properties of InsertResultTemplate data type 52](#_Toc271208988)

[Table 30: Properties of ResultTemplate data type 53](#_Toc271208989)

[Table 31: Properties of InsertResultTemplateResponse data type 54](#_Toc271208990)

[Table 32: Properties of InsertResult data type 57](#_Toc271208991)

[Table 33: Properties of GetResultTemplate data type 61](#_Toc271208992)

[Table 34: Properties of GetResultTemplateResponse data type 61](#_Toc271208993)

[Table 35: Properties of GetResult data type 64](#_Toc271208994)

[Table 36: Properties of GetResultResponse data type 65](#_Toc271208995)

[Table 37: XML schema implementation of types defined by this standard 69](#_Toc271208996)

[Table 38: DescribeSensor request KVP encoding 73](#_Toc271208997)

[Table 39: GetObservation request KVP encoding 74](#_Toc271208998)

[Table 40: GetFeatureOfInterest request KVP encoding 76](#_Toc271208999)

[Table 41: Action URIs for SOS message facets 79](#_Toc271209000)

1. Preface

This document specifies the interface of the Sensor Observation Service, hereinafter *SOS*. The SOS is one of a family of specifications that make up the OGC Sensor Web Enablement framework of specifications [OGC 06-021r4], hereinafter *SWE*. The functionality of the SOS within SWE is to provide standardized access to measured sensor observations as well as sensor descriptions. To encode observations the Observations & Measurements (O&M) standard is used. To encode sensor descriptions the Sensor Model Language (SensorML) is used. The SOS 2.0 builds on the previous SOS 1.0 efforts.

Suggested additions, changes, and comments on this document are welcome and encouraged. Such suggestions may be submitted by email message, or by making suggested changes in an edited copy of this document.

This standard deprecates previous versions, including version 1.0 [OGC 06-009r6].

Changes to Previous Version

This SOS 2.0 standard improves SOS 1.0 [OGC 06-009r6] by incorporating the following major changes:

* Reduced complexity through clearer structuring in *Core*, *Extensions* and *Profiles*.
* By applying the modular specifications policy [OGC 08-131r3], the implementation of the standard is facilitated. Clear listings of requirements and derived conformance tests also provide better support for CITE testing.
* Increased interoperability through restriction:
  + Certain operators and operands for temporal and spatial filters of *GetObservation*, *GetFeatureOfInterest* and *GetResult* have to be supported as a minimum by each SOS server which supports temporal or spatial filters.
  + The “Spatial Filtering Profile” considerably improves the spatial filter of observations in the *GetObservation* and *GetResult* operations.
* KVP binding: SOS 2.0 defines a simple key-value-pair binding for HTTP GET.
* SOAP binding: SOS 2.0 defines a SOAP binding for all its operations.
* Alignment with other SWE standards by relying on and reusing data types defined in the SWE Service Model [OGC 09-001].
* Improved concept for observation offerings: SOS 2.0 defines that each observation offering is limited to be associated with only one single sensor (system). This solves the ambiguity of SOS 1.0 of grouping observations to offerings.
* Improved handling of information on hosted observation offerings through application of *property inheritance mechanism* (defined by OGC 09-001) to decrease the amount of redundant information.
* The Capabilities document of SOS 2.0 lists *related features* instead of all *features of interest*. The related features are selected by the service provider and serve discovery purposes.
* The parameterization of the *GetFeatureOfInterest* operation is extended. Procedure, observed property and spatial operators are now allowed as filter parameters
* The handling of observation results is revised and extended. A new operation *InsertResult* now also allows the efficient upload of plain results.
* The operations *DescribeObservationType*, *DescribeResultType*, and *DescribeFeatureType* have been removed since the types of features, observations and results are identified by URIs in SOS 2.0 and these URIs should be known and could be resolvable.

Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 06-121r3], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

1. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

1. 52° North Initiative for Geospatial Open Source Software GmbH
2. University of Muenster – Institute for Geoinformatics (IfGI)
3. International Geospatial Services Institute GmbH (iGSI)
4. …
5. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

|  |  |  |
| --- | --- | --- |
| CONTACT | COMPANY | EMAIL |
| Arne Broering | 52°North | broering@52north.org |
| Christoph Stasch | University of Muenster | staschc@uni-muenster.de |
| Johannes Echterhoff | iGSI | johannes.echterhoff@igsi.eu |
| Peter Taylor | CSIRO | peter.taylor@csiro.au |
| Luis Bermudez | OGC | lbermudez@opengeospatial.org |
| Alex Robin | Spot Image | alexandre.robin@spotimage.fr |
| Jeff de La Beaujardière | NOAA | jeff.deLaBeaujardiere@noaa.gov |
| Tom Ingold | SAIC | thomas.b.ingold@saic.com |

1. Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Editor | Primary clauses modified | Description |
| 24.08.2009 | 0.1.0 | Arne Broering / Christoph Stasch | throughout | initial draft |
| 24.09.2009 | 0.2.0 | Arne Broering / Christoph Stasch | throughout | some modifications |
| 02.11.2009 | 0.3.0 | Christoph Stasch | throughout | some modifications |
| 04.11.2009 | 0.4.0 | Arne Broering | throughout | editorial changes |
| 19.07.2010 | 0.5.0 | Arne Broering | throughout | Merging of SOS specification parts to this doc.  General revision and editing.  “SOS Model Overview” included. |
| 20.07.2010 | 0.6.0 | Arne Broering | throughout | Deletion of GetDataAvailability (goes into external extension)  General restructuring |
| 26.07.2010 | 0.6.1 | Christoph Stasch | throughout | Minor corrections and comments. |
| 28.07.2010 | 0.7 | Arne Broering | throughout | Editorial changes (e.g., requirements tables)  Data type tables updated |
| 29.07.2010 | 0.7.1 | Arne Broering | throughout | GetFeatureOfInterest KVP Binding |
| 01.08.2010 | 0.8 | Christoph Stasch | 8.1 and 13 | FilterCapabilities section added  Spatial Filtering Profile added |
| 03.08.2010 | 0.9 | Christoph Stasch | Annex A | Abstract test suite added |
| 04.08.2010 | 0.9.1 | Arne Broering | throughout | general revision |
| 06.08.2010 | 0.9.2 | Christoph Stasch | Annex A | Tests for bindings added |
| 16.08.2010 | 0.9.3 | Peter Taylor | throughout | General revision. |
| 17.08.2010 | 0.9.4 | Arne Broering | throughout | Removed DescribeXXXType operations  Revised GetObservationById operation |
| 24.08.2010 | 0.9.5 | Arne Broering | throughout | Updated references to [OGC 09-001] |
| 02.09.2010 | 0.9.6 | Christoph Stasch, Johannes Echterhoff | throughout | general revision |

1. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

1. Future work

This version of the SOS interface defines a SOAP binding for all specified operations as well as a KVP binding for the core operations and the *GetFeatureOfInterest* operation. Future versions or extensions of this standard may add a RESTful binding similar to what has been defined by Janowicz et al. (2010)[[1]](#footnote-1).

Foreword

This version of the SOS standard supersedes and replaces OGC Implementation Specification [OGC 06-009r6].

*Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.*

*Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.*

Introduction

The Sensor Observation Service (SOS) provides a standardized interface for managing and retrieving metadata and observations from heterogeneous sensor systems. Sensor systems contribute the largest part of geospatial data used in geospatial systems today. Sensor systems include for example in-situ sensors (e.g. river gauges), moving sensor platforms (e.g. satellites or Autonomous Unmanned Vehicles) or networks of static sensors (e.g. seismic arrays). Used in conjunction with other OGC specifications the SOS provides a broad range of interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms, or networked constellations of sensors in real-time, archived or simulated environments.

The SOS is part of the Sensor Web Enablement (SWE) framework of standards [OGC 06-021r4]. The SWE activity aims at providing interfaces and protocols for enabling “Sensor Webs” through which applications and services are able to access sensors of all types. Sensor Webs can be accessed over networks such as the Internet with the same standard technologies and protocols that enable the Web.

SOS 2.0 relies on the Observations and Measurements (O&M) [ISO 19156:2010] standard to encode data gathered by sensors.

OGC® SOS 2.0 Interface Standard

# Scope

This OpenGIS**®** document specifies the interface standard of the Sensor Observation Service 2.0 (SOS). Every implementation of an SOS 2.0 shall adhere to this specification.

An SOS server can provide access to observations, sensor descriptions, as well as computational representations of observed features in an interoperable and standardized way. Further, an SOS server can provide means to insert new sensor descriptions or observations.

# Compliance

Standardization target are SOS 2.0 implementations (*Web Applications*, so *Web Servers* and *Web Clients*).

The following tables list the requirements classes defined by this standard.

Annex A lists the conformance tests which shall be exercised on any software artifact claiming to adhere to the SOS 2.0 standard.

Table : Main Requirements classes

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| Core | <http://www.opengis.net/spec/SOS/2.0/req/core> | The server implements the following operations: *GetCapabilities*, *DescribeSensor*, and *GetObservation*. |

Table : Requirements classes in Transactional Extension

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| Insertion Capabilities | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | The server provides metadata on its capabilities regarding the insertion of new sensors/observations. |
| Sensor Insertion | http://www.opengis.net/spec/SOS/2.0/req/sensorInsertion | The server implements the InsertSensor operation. |
| Sensor Deletion | http://www.opengis.net/spec/SOS/2.0/req/sensorDeletion | The server implements the DeleteSensor operation. |
| Observation Insertion | http://www.opengis.net/spec/SOS/2.0/req/obsInsertion | The server implements the InsertObservation operation. |

Table : Requirements classes in Result Handling in Extension

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| Result Insertion | http://www.opengis.net/spec/SOS/2.0/req/resultInsertion | The server implements the operations InsertResult and InsertResultTemplate. |
| Result Retrieval | http://www.opengis.net/spec/SOS/2.0/req/resultRetrieval | The server implements the operations GetResult and GetResultTemplate. |

Table : Requirements classes in Enhanced Operations Extension

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| Observation Retrieval By Id | http://www.opengis.net/spec/SOS/2.0/req/observationRetrievalById | The server implements the GetObservationById operation. |
| FeatureOfInterest Retrieval | http://www.opengis.net/spec/SOS/2.0/req/gfoi | The server implements the GetFeatureOfInterest operation. |

Table : Requirements classes in Spatial Filtering Profile

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| Spatial Filtering Profile | http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile | The server implements the Spatial Filtering Profile to improve spatial filtering of observations in GetObservation and GetResult (if supported) operations. |

Table : Requirements classes in Binding Extension

|  |  |  |
| --- | --- | --- |
| **Requirements class name** | **Requirements class identifier** | **Operation or behavior** |
| XML Encoding | http://www.opengis.net/spec/SOS/2.0/req/xml | The server encodes the data types from the conceptual model in XML as defined by this standard. |
| SOAP Binding | http://www.opengis.net/spec/SOS/2.0/req/soap | The server wraps XML encoded requests and results within SOAP Envelopes for the operations it supports. |
| Core KVP Binding | http://www.opengis.net/spec/SOS/2.0/req/kvp-core | The server implements the key-value pair encoding for the operations *GetCapabilities*, *DescribeSensor*, and *GetObservation*. |
| GetFeatureOfInterest KVP Binding | http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi | The server implements the key-value pair encoding for the operation *GetFeatureOfInterest*. |

# Normative references

This *OGCSOS 2.0* specification consists of the present document and an XML Schema. The complete specification is identified by OGC URI <http://www.opengis.net/spec/SOS/2.0>, the document has OGC URI <http://www.opengis.net/doc/IS/SOS/2.0>.

The complete specification is available for download from <http://www.opengeospatial.org/standards/sos> In addition, the XML Schema is posted online at <http://schemas.opengis.net/sos/2.0> as part of the OGC schema repository. In the event of a discrepancy between bundled and schema repository versions of the XML Schema files, the schema repository is considered authoritative.

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*

ISO 8601:2000, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 19107:2003, *Geographic information — Spatial schema*

ISO 19108:2002, *Geographic information — Temporal schema*

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

ISO 19143:2010, *Geographic information — Filter Encoding*

ISO 19156:2010, *Geographic information — Observations and Measurements*

OGC® Policy Standard, *The Specification Model - A Standard for Modular specifications*, Version 1.0.0, OGC document 08-131r3

OpenGIS® Encoding Standard, *Geography Markup Language*, Version 3.2.1, OGC document 07-036

OpenGIS® Encoding Standard, *SWE Common Data Model*, Version 2.0, OGC document 08-094

OpenGIS® Encoding Standard, *SensorML Version 1.0 Schema - Corrigendum 1*, Version 1.0.1, OGC document 07-022r2

OpenGIS® Implementation Standard, *Observations and Measurements - XML Implementation*, Version 2.0, OGC document 10-025

OpenGIS® Implementation Standard, *SWE Service Model*, Version 2.0, OGC document 09-001[[2]](#footnote-2)

OpenGIS® Implementation Standard, *Web Services Common*, Version 1.1.0, OGC document 06-121r3

OpenGIS® Best Practice, *OGC® Sensor Web Enablement Architecture*, Version 0.4, OGC document 06-021r4

# Terms and definitions

For the purposes of this standard, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] and Clause 3 of the SWE Architecture document [OGC 06-021r4] shall apply. In addition, the following terms and definitions apply.

## Measurement

An observation whose result is a measure. [OGC 07-022r1]

## Observable

Parameter or a characteristic of a phenomenon subject to observation. [OGC 07-022r1]

## Observation Offering

An Observation Offering groups collections of observations produced by onesensor system and lists the basic metadata for the associated observations including the procedure which made the observations.

## Procedure

Method, algorithm , instruments, sensor and sensor systems (e.g. platforms). [OGC 07-022r1]

## Result

An estimate of the value of some property generated by a known procedure. [OGC 07-022r1]

# Conventions

## Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 06-121r3] apply to this document, plus the following abbreviated terms.

GML Geography Markup Language

ISO International Organization for Standardization

OGC Open Geospatial Consortium

OWS OGC Web Services

O&M Observations and Measurements

SensorML Sensor Model Language

SOS Sensor Observation Service

SWE Sensor Web Enablement

UML Unified Modeling Language

XML eXtensible Markup Language

## UML notation

Diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

NOTE Packages and data types from foreign namespaces are shown with grey background.

## Platform-neutral and platform-specific standards

For compliance with Clause 10 of OGC Topic 12 and ISO 19119, this specification follows the pattern defined in Subclause 5.4 of [OGC 06-121r3]. That is, model elements are specified in platform-neutral fashion first, using tables that serve as data dictionaries for the UML model (see Subclause 5.4 of this document). Platform-specific encodings of these model elements are provided in separate clauses or documents.

The XML Schema encoding has automatically been generated using the mapping rules defined in OGC 09-001.

This document specifies a platform-specific encoding for a SOAP operation binding and a KVP binding over HTTP. The model as well as its XML Schema encoding (and other data) can also be used to create other bindings such as a REST(ful) or POX (Plain Old XML) over HTTP (using plain XML).

## Data dictionary tables

The UML model data dictionary is specified herein in a series of tables. The contents of the columns in these tables are described in Table 1 of [OGC 06-121r3]. The contents of these data dictionary tables are normative, including any table footnotes.

## Classes imported from other specifications with predefined XML encoding

This specification uses an automatic mapping approach from the UML model to the XML Schema encoding. The approach is described in OGC 09-001. This standard uses types defined by other standards. For the mapping to XML Schema, the implementation instructions listed in table D.2 of [OGC 07-036] are used together with the instructions listed in to in this standard.

Some of the properties defined in the conceptual model of this standard which point to objects rather than directly containing them are encoded as described in Subclause 24.2.4.11 of [OGC 09-001]. An XML Schema implementation for these types is therefore not needed in this section.

For an explanation of the table columns, see Subclause D.2.1 of [OGC 07-036].

Table : XML Schema implementation of types from the SWE Service Model [OGC 09-001]

|  |  |  |  |
| --- | --- | --- | --- |
| **UML class** | **object element** | **type** | **property type** |
| AbstractContents | swes:AbstractContents | swes:AbstractContentsType | swes:AbstractContentsPropertyType |
| AbstractOffering | swes:AbstractOffering | swes:AbstractOfferingType | swes:AbstractOfferingPropertyType |
| ExtensibleRequest | swes:ExtensibleRequest | swes:ExtensibleRequestType | swes:ExtensibleRequestPropertyType |
| ExtensibleResponse | swes:ExtensibleResponse | swes:ExtensibleResponseType | swes:ExtensibleResponsePropertyType |
| InsertionMetadata | swes:InsertionMetadata | swes:InsertionMetadataType | swes:InsertionMetadataPropertyType |

Table : Implementation of types from OWS Common [OGC 06-121r3]

|  |  |  |  |
| --- | --- | --- | --- |
| **UML class** | **object element** | **type** | **property type** |
| GetCapabilities | - | ows:GetCapabilitiesType | - |
| OWSServiceMetadata | - | ows:CapabilitiesBaseType | - |

Table : Implementation of types from SWE Common Data Model [OGC 08-094]

|  |  |  |  |
| --- | --- | --- | --- |
| **UML class** | **object element** | **type** | **property type** |
| AbstractDataComponent | swe:AbstractDataComponent | swe:AbstractDataComponentType | swe:AbstractDataComponentPropertyType |
| AbstractEncoding | swe:AbstractEncoding | swe:AbstractEncodingType | swe:AbstractEncodingPropertyType |

Table : Implementation of types from ISO 19143 (Filter Encoding Specification)

|  |  |  |  |
| --- | --- | --- | --- |
| **UML class** | **object element** | **type** | **property type** |
| Filter\_Capabilities | fes:Filter\_Capabilities | - | - |
| SpatialOperator | fes:spatialOps | fes:SpatialOpsType | - |
| TemporalOperator | fes:temporalOps | fes:TemporalOpsType | - |

## Namespace Conventions

This standard uses a number of namespace prefixes throughout; they are listed in . Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table : Prefixes and Namespaces used in this standard

|  |  |
| --- | --- |
| **Prefix** | **Namespace** |
| fes | http://www.opengis.net/fes/2.0 |
| gml | http://www.opengis.net/gml/3.2 |
| om | http://www.opengis.net/om/2.0 |
| ows | http://www.opengis.net/ows/1.1 |
| soap11 | http://schemas.xmlsoap.org/soap/ |
| soap12 | http://www.w3.org/2003/05/soap-envelope |
| sos | http://www.opengis.net/sos/2.0 |
| swe | http://www.opengis.net/swe/2.0 |
| swes | http://www.opengis.net/swes/2.0 |
| wsa | http://www.w3.org/2005/08/addressing |
| xs | http://www.w3.org/2001/XMLSchema |

# Observation Model Overview

SOS is primarily designed to provide access to observations. The model for Observations and Measurements (O&M) is defined in the O&M standard [ISO 19156]. An XML encoding of this conceptual model is defined in the OGC implementation specification [OGC 10-025].

The basic Observation model is depicted in .



Figure : O&M Basic Observation Model

An Observation provides a *result* whose value is an estimate of a property of the observation target, the *feature of interest*; i.e. an observation is a property-value-provider for the feature of interest. An instance of an Observation is classified by its *phenomenonTime*, *featureOfInterest*, *observedProperty*, and the *procedure* used. The procedure is usually a sensor but can also be for example a computation or post-processing step. More detailed information about the observation data model can be found in [ISO 19156].

# SOS Model Overview

As shown in , the *Core* of the SOS 2.0 defined in Clause builds upon certain specifications. Based on the Core, extensions can be defined to add further functionality. This document contains the following extensions, the *Transactional Extension* (Clause ), *Result Handling* *Extension* (Clause ), the *Enhanced Operations Extension* (Clause ), and the *Binding Extension* (Clause ). Further, this document contains the profile *Spatial Filtering Profile* (Clause ).

Future extensions may specify additional functionality. Future profiles, in particular for certain domains (e.g. hydrology or oceanography) may further restrict the SOS to increase and facilitate interoperability for their use cases.

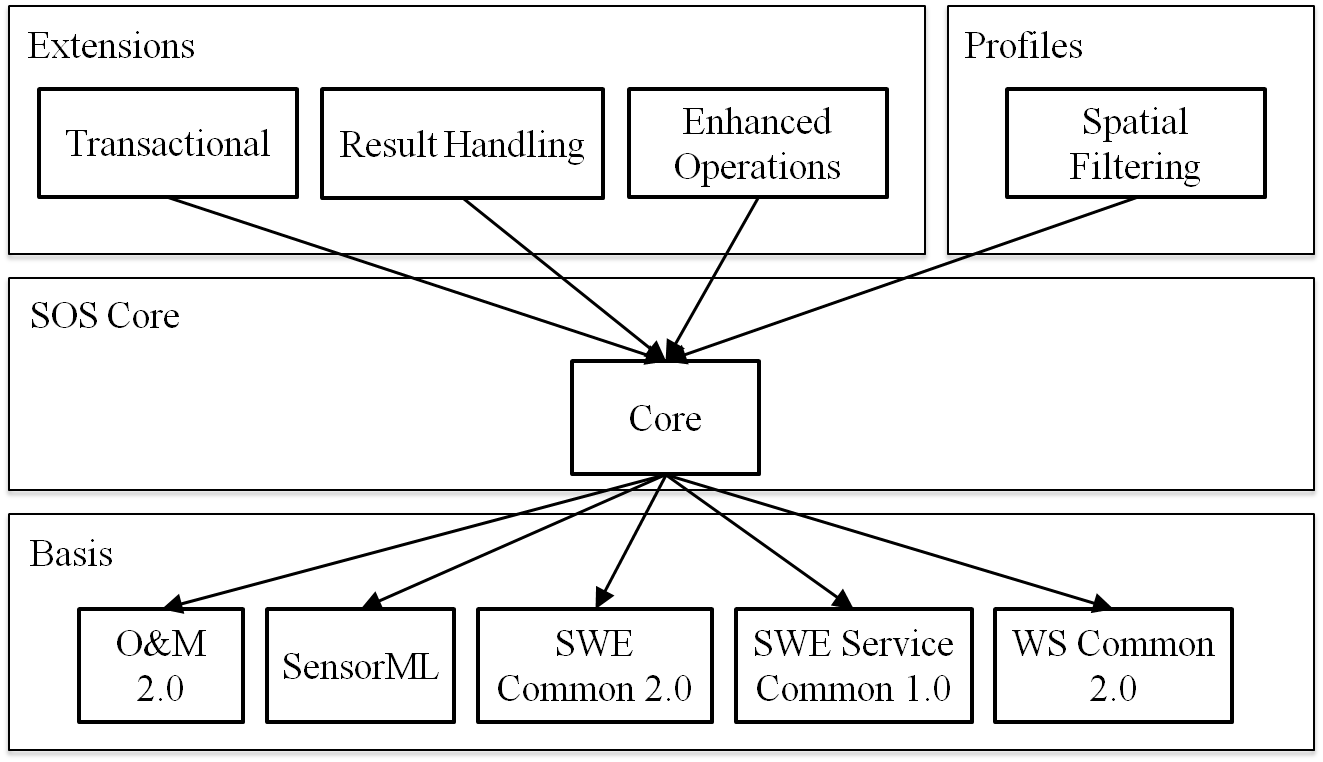


Figure : SOS core, extensions and OGC/SWE specification basis

The SOS Core defines three operations:

***GetCapabilities*** - provides access to metadata and detailed information about the operations available by an SOS server.

***DescribeSensor*** - provides access to detailed information about the sensors and sensor systems available by an SOS server.

***GetObservation*** - provides access to sensor and sensors system observations by allowing spatial, temporal and thematic filtering.

Further, this document defines extensions of the SOS core which specify additional operations as listed below.

Enhanced Operations Extension:

***GetObservationByID -*** provides access to observations from an SOS by passing only the ID of an observation.

***GetFeatureOfInterest* -** provides access to the features of interest for which the SOS offers observations.

Transactional Extension:

***InsertSensor*** – allows registration of new sensors at the SOS.

***DeleteSensor*** – allows the deletion of registered sensors and all their associated observations.

***InsertObservation* –** allows the insertion of observations in an SOS server.

Result Handling Extension:

***InsertResult*** - allows the insertion of observation results in an SOS server. Before inserting, it is necessary that a template with observation metadata exists in the server.

***InsertResultTemplate* -** allows the insertion of an observation template, containing the observation metadata and structure of the results. This operation is necessary for later insertion of observation results.

***GetResultTemplate* -** provides access to a template containing the structure of results returned by later invocation of the GetResult operation.

***GetResult -*** provides access to an observation result without the observation metadata and without the information about the structure of the results.

The following sections describe the interaction with the SOS to retrieve and insert observations, and to retrieve and insert results (no observation metadata). The operations to support these interactions are defined in the Core and extensions of this document.

## Workflow of Observation Retrieval

and the following descriptions illustrate the sequence of operation calls to obtain and interpret observations from an SOS.

1. To obtain an up-to date listing of available data a client issues a *GetCapabilities* request to the server.
2. It may then issue a *DescribeSensor* request to find out further details about particular procedures associated with the SOS.
3. The client may also call the *GetFeatureOfInterest* operation to get the detailed description of a particular feature, or to get a list of all features of interest for specified spatial filters, observed properties or procedures. The information returned by the *GetFeatureOfInterest* operation may be used by the client to choose appropriate parameters for the *GetObservation* call.
4. The client issues the *GetObservation* request and retrieves the observations.



Figure : Operation sequence for observation retrieval

## Workflow of Observation Insertion

As illustrated in , a sensor data producer first requests the service metadata before new observations can be inserted into an SOS.

1. The service metadata is retrieved by invoking the *GetCapabilities* operation (Subclause ). The returned Capabilities document contains, within the *Contents* section (Subclause ), the procedures which are registered at the SOS. The *InsertionCapabilities* section (Clause ) lists the observation types and result types which are supported by the SOS for insertion.
2. If it has not been registered at the SOS, the *InsertSensor* operation (Subclause ) is called to insert the procedure description of the observation procedure. The SOS returns the *ObservationOffering* to which the procedure has been assigned.
3. Finally, the observation can be inserted using the *InsertObservation* operation, by specifying the *ObservationOffering* to which it shall be uploaded.



Figure : Operation sequence for observation insertion

## Workflow of Result Insertion

An SOS may support the *Result Insertion* requirements class. It contains two operations, *InsertResultTemplate* (Subclause ) and *InsertResult* (Subclause ), which allow inserting sensor results into an SOS without the need to repeatedly transmit the complete set of observation metadata. The operations can be used, if the metadata contained in the produced observations remain the same. This is useful if the communication bandwidth and processing power of the sensor data producer is limited. depicts the sequence of operation calls for the insertion of results into an SOS.

1. The service metadata is retrieved by invoking the *GetCapabilities* operation (Subclause ). The returned Capabilities document contains, within the *Contents* section (Subclause ), the procedures which are registered at the SOS. The *InsertionCapabilities* section (Clause ) lists the observation types and result types which are supported by the SOS for insertion.
2. If it has not been registered at the SOS, the *InsertSensor* operation (Subclause ) is called to insert the procedure description of the observation procedure. The SOS returns the *ObservationOffering* to which the procedure has been assigned.
3. A result template is sent to the SOS by using the *InsertResultTemplate* operation. This result template contains observation metadata for the results, the *ObservationOffering* to which the data shall be uploaded, and a structural description of the results.
4. The sensor data producer inserts the results without sending the whole observation metadata and by referencing the inserted result template.



Figure : Sequence of operations for insertion of results

## Workflow of Result Retrieval

An SOS may support the *Result Retrieval* requirements class. It contains two operations, *GetResultTemplate* (Subclause ) and *GetResult* (Subclause ), for the retrieval of observation results without the complete set of observation metadata. The purpose of the Result Retrieval requirements class is to allow clients to repeatedly obtain sensor data without having to send and receive requests and responses that largely contain the same data except for a new timestamp. A client can repeatedly request sensor data from one or more sensors on a recurring basis. This is in particular useful in scenarios with restricted bandwidth or clients with restricted processing power. and the following descriptions illustrate the operation sequence for result retrieval.

1. The client invokes the *GetResultTemplate* operation to retrieve the resultStructure and the resultEncoding which will be used by the SOS in later *GetResult* operation responses.
2. The client calls the *GetResult* operation and retrieves result values.



Figure : Sequence diagram of result retrieval

# Requirements Class: Core

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/core** | |
| **Target Type** | Web Application |
| **Dependency** | urn:iso:ts:iso:19156:clause:6 |
| **Dependency** | http://www.opengis.net/doc/IS/OWS/1.1/clause/7 |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/7 |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/9 |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/15 |

In this clause, the SOS core operations *GetCapabilities*, *DescribeSensor*, and *GetObservation* are specified.

***GetCapabilities*** - provides access to metadata and detailed information about the operations available by an SOS server.

***DescribeSensor*** - provides access to detailed information about the sensors and sensor systems available by an SOS server.

***GetObservation*** - provides access to sensor and sensors system observations by allowing spatial, temporal and thematic filtering.

The SOS operations follow the general pattern of other OGC Web Services and where appropriate inherit or re-use elements defined by the OWS Common standard [OGC 06-121r3] and the SWE Service Model standard [OGC 09-001].

All request and response types defined in this standard (except the *GetCapabilities* request and response which are based on [OGC 06-121r3]) are derived from the *ExtensibleRequest* or *ExtensibleRepsonse* type, respectively, as defined in OGC 09-001. Each request and response type defines an optional extension property, a container for request parameters that can be defined by an extension.

This standard defines the following general requirements applying to **ALL** operations defined in the following (except where explicitly negated):

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/request-service** |
| 1. For **ALL** SOS request types defined in this standard, a mandatory service parameter specifies the OWS type abbreviation of the implementing service. It is of type CharacterString and shall have the fixed value “SOS”. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/request-version** |
| 1. For **ALL** SOS request types defined in this standard except the request type of the GetCapabilities operation, a mandatory version parameter specifies the service type specification. It is of type CharacterString and shall have the fixed value “2.0.0”. |

## GetCapabilities Operation

This operation allows clients to retrieve the service metadata (also called the “Capabilities” document) of an SOS server.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc** |
| 1. Every SOS server shall support the *GetCapabilities* operation as defined in this Clause. |

The conceptual model of the *GetCapabilities* operation is shown in the following UML diagram.

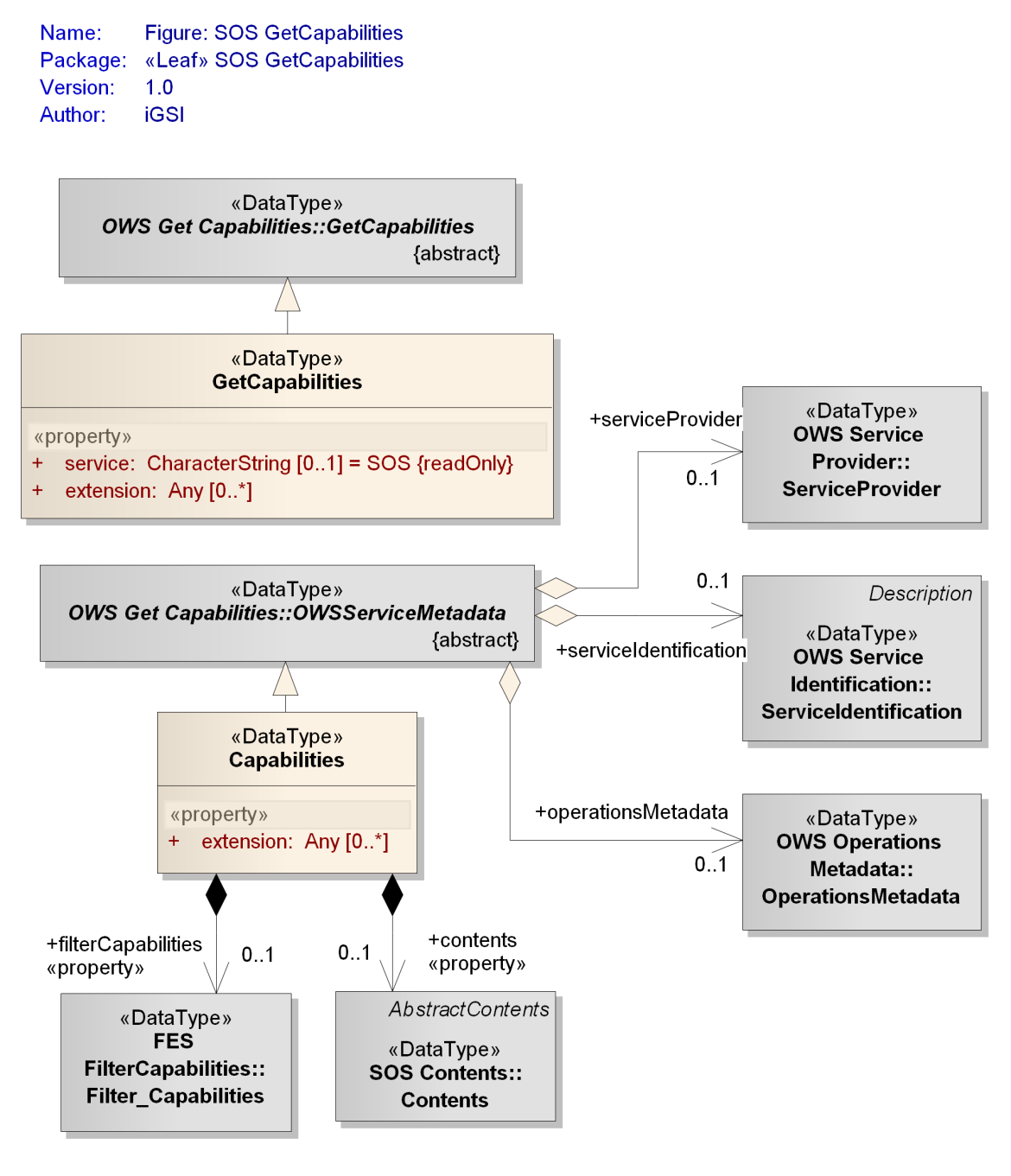


Figure : Data types of the GetCapabilities operation

### Request

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-ows** |
| 1. The *GetCapabilities* operation request shall be implemented as specified in Clause 7 of OWS Common [OGC 06-121r3]. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-version** |
| 1. If the AcceptVersions parameter is contained in the request, it shall contain the character string “2.0.0”. |

The SOS *GetCapabilities* data type derives from the OWS Common *GetCapabilities* data type (listed in table 3 of [06-121r3]) and thus inherits all the properties contained in that data type.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-request-structure** |
| 1. In addition to the properties inherited from OWS Common *GetCapabilities*, the SOS *GetCapabilities* shall include the properties according to |

Table : Properties in the GetCapabilities data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| extension | container for elements defined by extension specifications | Any type  value is defined by the extension specification | Zero or more (optional) |
| service | service type identifier | Character String type, not empty  value shall be “SOS” | Zero or one (optional)  default value is “SOS” |

Note: the request property – derived from OWS Common GetCapabilities type – is explicit or implied by each specific binding of the GetCapabilities operation, so is not necessarily part of the request representation defined by that binding.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-sections** |
| 1. The allowed set of values for the *sections* parameter shall be as specified in Table 10 of [OGC 06-121r3], with the additions listed in below. |

Note: additional sections can be added by extensions; this is done for example by the InsertionCapabilities requirements class (clause ).

Table : Additional section names for SOS Capabilities

|  |  |
| --- | --- |
| **Section name** | **Meaning** |
| Contents | The Contents section of the SOS service metadata document contains information about the data offered by the SOS server (Subclause ). |
| FilterCapabilities | The FilterCapabilties section of the SOS service metadata document contains information about the supported filters (Subclause ) |

### Response

The response to a *GetCapabilities* operation request is also called the Capabilities document. This document provides clients with service metadata about a specific service instance, including metadata about the data served.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-response** |
| 1. The response to a successful *GetCapabilities* request shall return an instance of the SOS *Capabilities* type. |

The SOS *Capabilities* data type derives from the OWS Common *OWSServiceMetadata* data type (as defined in clause 7.4.2 of [06-121r3]) and thus inherits all the properties contained in that data type.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-structure** |
| 1. In addition to the properties inherited from OWS Common *OWSServiceMetadata*, the SOS *Capabilities* shall include the properties according to |

Table : Properties in the Capabilities data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| contents | metadata about the observations and procedures hosted by the service | Contents, see | Zero or one (optional)  inclusion depends on the values in the Sections parameter of the GetCapabilities operation request |
| extension | container for elements defined by extension specifications | Any type  value is defined by the extension specification | Zero or more (optional) |
| filterCapabilities | metadata about the supported filter functionality | Filter\_Capabilities, see ISO 19143 | Zero or one (optional)  inclusion depends on the values in the Sections parameter of the GetCapabilities operation request |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-version** |
| 1. The default version of the Capabilities document returned by a service implementing this standard shall be “2.0.0”. |

The ServiceIdentification, ServiceProvider, and OperationsMetadata sections of the *GetCapabilities* response document are defined in [OGC 06-121r3].

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-conf-class-listing** |
| 1. The Capabilities document shall advertise conformance classes which are supported by the server. Therefore, each value of the Profile property of the ServiceIdentification section shall be the pointer to a conformance class, and the server shall pass all tests defined for each listed conformance class. |

NOTE Typically, such conformance classes will be specified in SOS extensions.

The OperationsMetadata section specified in [OGC 06-121r3] lists the request types supported by an SOS server. For the SOS Core, these are *GetCapabilities*, *DescribeSensor*, and *GetObservation*; extensions may add further request types to this list.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-operation-listing** |
| 1. In the response to a successful GetCapabilities request, the Operations­Meta­data section shall contain three Operation elements with case-sensitive name values “GetCapabilities”, “DescribeSensor”, and “GetObservation”. All other operations supported by the SOS server and defined in extensions shall be listed in the same way. |

The following subclauses define the sections of the SOS Capabilities document which are added by this standard, the FilterCapabilities and the Contents section.

#### FilterCapabilities Section

The FilterCapabilities section is imported from [ISO 19143] and used to state which filter operators and operands are supported by an SOS server. The operators and operands refer to the parameters of service operations that include OGC filter expressions, like the *GetObservation* operation.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-spatialFilter-listing** |
| 1. In the response to a successful GetCapabilities request, the FilterCapabilities section shall list the spatial filter operators and operands which are supported by the service as defined in section 7.14.5 of [ISO 19143]. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/spatial-filter-minimum** |
| 1. Each SOS shall at least support the spatial filter operator BBOX. |

1. An SOS which supports the spatial operator Intersects for the geometry operands gml:Point and gml:Polygon shall list these as shown in the example at <http://test.schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1_response.xml>

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-temporalFilter-listing** |
| 1. In the response to a successful GetCapabilities request, the FilterCapabilities section shall list the temporal filter operators and operands which are supported by the service as defined in section 7.14.6 of [ISO 19143]. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/temporal-filter-minimum** |
| 1. Each SOS shall at least support the following temporal filter operators:    1. TEquals operator in conjunction with the TimeInstant type.    2. TDuring operator in conjunction with the TimePeriod type. |

1. An SOS which supports the temporal operator During for the geometry operands gml:TimeInstant and gml:TimePeriod shall list these as shown in the example at <http://test.schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1_response.xml>

#### Contents Section

The Contents section of the *GetCapabilities* response describes the data offered by an SOS server. To group the offered observations the SOS defines the concept of *ObservationOfferings*.

The Contents type lists all ObservationOfferings of an SOS server. An ObservationOffering groups collections of observations produced by **one** sensor system[[3]](#footnote-3). The ObservationOffering lists the basic metadata for the associated observations including the procedure which made the observations. An observation may belong to more than one ObservationOffering.

To summarize: there is a 1:n relationship between procedures and ObservationOfferings; there is a n:m relationship between observations created by these procedures and ObservationOfferings

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-offerings-observations** |
| 1. The observations that an offering provides information about shall not have been created by a procedure other than the one that is stated by the offering. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-contents** |
| 1. If requested by a client, the Capabilities document of an SOS shall contain a section of type Contents. |

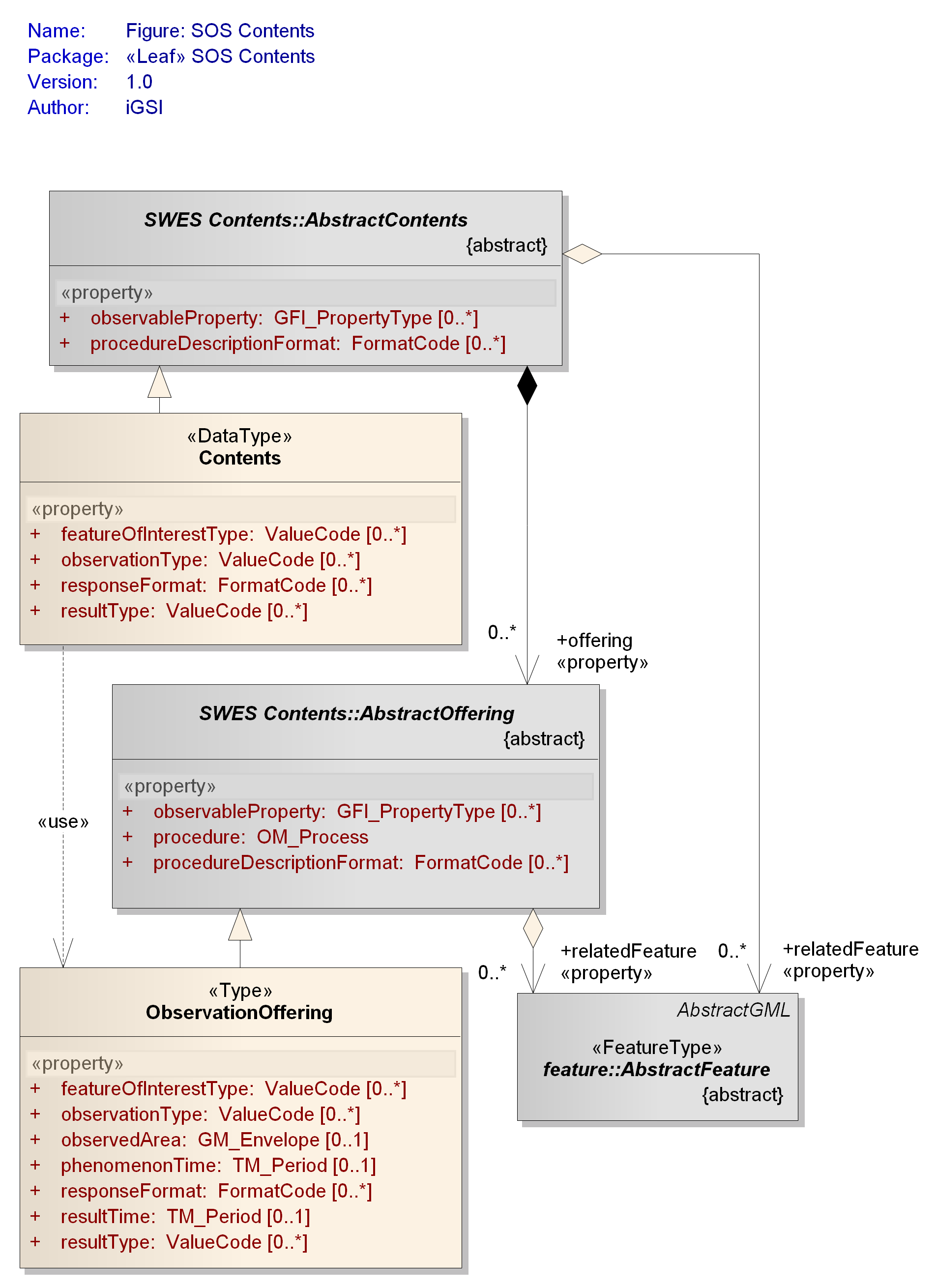


Figure : Data types of Contents section

The SOS Contents type derives from the SWES AbstractContents type defined in **OGC 09-001 and inherits its properties.**

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-contents-structure** |
| 1. In addition to the properties inherited from SWES *AbstractContents*, the SOS *Contents* shall include the properties according to |

Table : Properties of Contents data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity** |
| responseFormat | Identifies the response format supported for observation retrieval. | FormatCode  see OGC 09-001 | Zero or many (optional) |
| featureOfInterestType | Identifies the type of features of interest associated with the observations provided by the service. | ValueCode  see clause | Zero or many (optional) |
| observationType | Identifies the type of observation which is used by the service to encode observations. | ValueCode  see clause | Zero or many[[4]](#footnote-4) (optional) |
| resultType | Identifies the type of result which is used by the service to encode results of observations (whose type definitions do not define a unique result type). | ValueCode  see clause | Zero or many4 (optional) |

The SOS ObservationOffering type derives from the SWES AbstractOffering type defined in OGC 09-001 and inherits all its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-observationoffering-structure** |
| 1. In addition to the properties inherited from SWES *AbstractOffering*, the SOS *ObservationOffering* shall include the properties according to |

Table : Properties of ObservationOffering data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| phenomenonTime | period of time that contains the phenomenon times of all observations belonging to the offering | TM\_Period  see [ISO 19108] | Zero or one (optional)  is omitted if the offering has no observations associated to it |
| resultTime | period of time that contains the result times of all observations belonging to the offering | TM\_Period  see [ISO 19108] | Zero or one (optional)  is omitted if the offering has no observations associated to it |
| observedArea | bounding box of the spatial extent of all features of interest that belong to observations associated with the offering | GM\_Envelope  see [ISO 19107] | Zero or one (optional) |
| responseFormat | *As defined in .* | | |
| featureOfInterestType | *As defined in .* | | |
| resultType | *As defined in .* | | |
| observationType | *As defined in .* | | |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-offering-identifier** |
| 1. The service shall assign a unique identifier value to each of its ObservationOfferings. |

For the purpose of reducing the amount of redundant information in the Capabilities document, the SOS supports a so-called *property inheritance mechanism* as defined in [OGC 09-001].

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-property-inheritance-mechanism** |
| 1. The ObservationOffering is a property inheritor and the Contents its property provider as defined by OGC 09-001.   Clients shall apply the SWES *property inheritance mechanism* according to the inheritance categories defined for the ObservationOffering properties in when determining their values (for a given offering).  The service shall ensure that the number of values it provides for the properties of each offering is as stated in when the *property inheritance mechanism* was applied. |

The following table shows which of the properties of an ObservationOffering can be inherited and which cardinality is expected after the inheritance mechanism has been applied.

Table : Inheritance of ObservationOffering properties (from Contents)

|  |  |  |
| --- | --- | --- |
| **Property** | **Number[[5]](#footnote-5)** | **Inheritance** |
| observableProperty | 1..\* | replace |
| procedure | 1 | no |
| procedureDescriptionFormat | 1..\* | replace |
| relatedFeature | 0..\* | replace |
| featureOfInterestType | 1..\* | replace |
| observationType | 1..\* | replace |
| observedArea | 0..1 | no |
| phenomenonTime | 0..1 | no |
| responseFormat | 1..\* | replace |
| resultTime | 0..1 | no |
| resultType | 0..\* | replace |

Thus, even though the UML model and schema encoding define for example the observationType and responseFormat properties as optional, they are mandatory in each ObservationOffering. In other words, each offering has to include at least one value for these two properties after the property inheritance mechanism was applied.

**As O&M v2.0 is the only required format for encoding observation data at a SOS:**

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-format-om20** |
| 1. The responseFormat of each ObservationOffering shall have at least the value *http://www.opengis.net/om/2.0*. |

Other response formats may be supported as well. However, the way how such formats are supported needs to be described in a specific extension to this document.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-observation-result-type** |
| 1. If one of the observation type identifiers listed in the observationType property of an ObservationOffering does not implicitly define the type of the observation result then the identifiers of the result types used by observations that are of these (observation) types and which are associated to the ObservationOffering shall be listed in the resultType property of that offering. |

### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/gc-exception** |
| 1. When an SOS server encounters an error while performing a GetCapabilities operation, it shall return an exception message encoded as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetCapabilities operation

* MissingParameterValue
* InvalidParameterValue
* VersionNegotiationFailed
* InvalidUpdateSequence
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

### Examples

1. An example request of the XML implementation of this operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1.xml>

1. An example response of the XML implementation of this operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1_response.xml>

## DescribeSensor Operation

The *DescribeSensor* operation enables the retrieval of metadata descriptions of procedures (or: *sensors*) associated with an SOS. While the response to a *GetCapabilities* request lists procedures associated with the SOS the *DescribeSensor* operation can be used subsequently to retrieve detailed definitions for the listed procedures. The definition of this operation can be found in Clause 11 of [OGC 09-001].

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/ds** |
| 1. Every SOS server shall support the *DescribeSensor* operation as defined in Clause 11 of [OGC 09-001]. When targetting procedures hosted by the SOS, the service attribute in the DescribeSensor request shall have the value “SOS” and the version attribute shall have the value “2.0.0”. |

### Examples

1. An example request of the XML implementation of this operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/DescribeSensor1.xml>

## GetObservation Operation

The *GetObservation* operation is designed to query an SOS to retrieve observation data structured according to the O&M specification. Other response formats are possible; however, O&M is the default and mandatory format for every SOS server. Several parameters of the *GetObservation* operation allow extensive filtering of the observations requested from the SOS server.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go** |
| 1. Every SOS server shall support the *GetObservation* operation as defined here. |

The conceptual model of the *GetObservation* operation is shown in the following UML diagram.

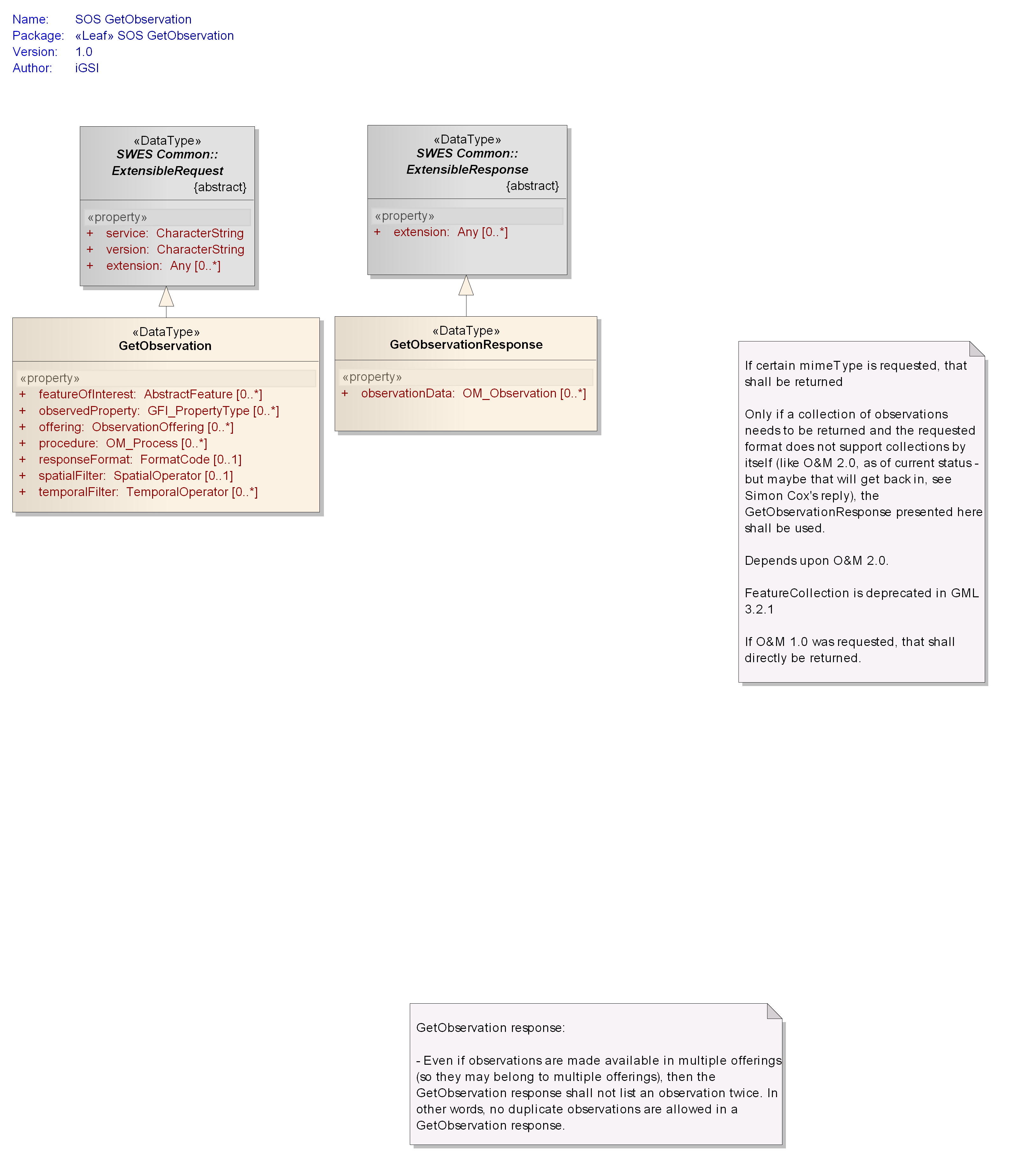


Figure : Data types of GetObservation operation

### Request

A GetObservation operation request contains parameters that constrain the observations to be retrieved from a SOS.

The GetObservation data type is derived from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS GetObservation operation request type shall include the properties according . |

The concrete representation of this structure depends on the chosen protocol binding.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-parameters** |
| 1. The SOS returns all observations that match the specified parameter values. The filter parameters (e.g., observedProperty, procedure, or temporalFilter) shall be connected with an explicit **AND**. The values of each of the parameters are connected with an implicit **OR**. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-omitting-parameters** |
| 1. If an optional parameter of a *GetObservation* request is not specified by the client, the filter (represented by the parameter) shall not be applied to the observation set which will be returned by the SOS server. |

For example, in consequence of , an SOS server returns observations of all time to the client if the temporal filter is omitted.

NOTE: An implementation of an SOS server may return an exception message as specified in Clause 15 of [OGC 09-001] if the response of a *GetObservation* request would be too big to be reasonably send to a client.

1. Resulting from and an abstract *GetObservation* request looks like this:

*GetObservation* ( featureOfInterest := *weatherstation\_in\_my\_backyard*

**AND** observedProperty := *temperature*

**AND** procedure := *thermometerX* **OR** *anemometerY*)

This request returns the observations of all offerings, all time and all spatial extent, and which were made for the feature of interest “weather\_station\_in\_my\_backyard”, and which carry results for the observed property “temperature” and were made by the sensor “thermometerX” or “anemometerY”.

Table : Properties of GetObservation request data type

| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| --- | --- | --- | --- |
| offering | Pointer to an ObservationOffering advertised in the Capabilities document for which observations are requested. | ObservationOffering id  see | Zero or many[[6]](#footnote-6) (Optional) |
| observedProperty | Pointer to an observedProperty for which observations are requested. | GFI\_PropertyType id  see [ISO 19156] | Zero or many (optional) |
| procedure | Pointer to a sensor system for which observations are requested. It defines a filter for the procedure property of the observations | OM\_Process id  see [ISO 19156] | Zero or many (Optional) |
| temporalFilter | Specifies a filter for a time property of requested observations. The supported time range for the phenomenonTime property of observations is listed in the selected ObservationOffering. The supported temporal operands and operators are listed in the FilterCapabilities section of the Capabilities document. | TemporalOperator  see [ISO 19143] | Zero or many (Optional) |
| spatialFilter | Specifies a filter[[7]](#footnote-7) which applies to a spatial property of an observation. This property is defined in the valueReference element of the SpatialOperator. | SpatialOperator  see [ISO 19143] | Zero or one (Optional) |
| featureOfInterest | Pointer to a feature of interest for which observations are requested. | AbstractFeature id  see clause D.3.4 in [ISO 19136] and clause 9.3 in [OGC 07-036] | Zero or many (Optional) |
| responseFormat | Identifier of desired responseFormat for the requested observations. The supported responseFormats are listed in the ObservationOffering. | FormatCode  see OGC 09-001  Default is O&M 2.0 [ISO 19156] identified by the value http://www.opengis.net/om/2.0 | Zero or one (optional)  If not provided, default value is assumed. |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

NOTE If every parameter in GetObservation operation is optional, empty queries can be submitted to the SOS. This might result in huge responses. An SOS server may solve this issue by returning an exception if a request would result in a too big response.

NOTE Observations returned by the SOS might be encoded with different observation types and result types. For the client, such responses are difficult to parse. This issue is not addressed here. It might be solved in future through an extension which allows requesting only certain observation / result types. However, such a mechanism would put the burden of observation / result type transformation on the SOS server.

### Response

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-default-response-format** |
| 1. An SOS server shall be able to return the requested observations encoded as O&M 2.0 [ISO 19156]. O&M 2.0 is the default response format for the *GetObservation* operation and shall be returned in the response if no other specific responseFormat was specified in the request. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-response-format** |
| 1. The SOS server shall respond in the specified responseformat. If the response format specified by the client is not supported an exception message shall be returned with the exception code “InvalidParameterValue” and the locator value ”responseFormat”. |

The SOS GetObservationResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-response-type** |
| 1. If O&M 2.0 [ISO 19156] is returned by an SOS server, the response shall contain an instance of the GetObservationResponse type   In addition to the properties inherited from SWES ExtensibleResponse, the SOS GetObservationResponse shall include the properties according to . |

An instance of GetObservationResponse type may be empty if none of the observations associated with the SOS fulfill the *GetObservation* parameters specified by the client. If the response contains observations, the observationData elements are of type OM\_Observation as defined in [ISO 19156] or subtypes of OM\_Observation.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-observation-duplicates** |
| 1. A response to a *GetObservation* request shall not contain observation duplicates. Even if multiple ObservationOfferings are requested and an observation belongs to more than one ObservationOffering, then the *GetObservation* response shall not list such an observation twice. |

Table : Properties of GetObservationResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| observationData | Observation which fulfills the *GetObservation* request. | OM\_Observation  (see [ISO 19156]) | Zero or many (optional) |

### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/go-exception** |
| 1. When an SOS server encounters an error while performing a GetObservation operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetObservation operation

* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

### Examples

1. An example request and response of the XML implementation of the *GetObservation* operation with a specified observedProperty and temporalFilter can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation1_obsProps.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation1_obsProps_response.xml>

1. An example request and response of the XML implementation of the *GetObservation* operation with a specified observedProperty and procedure can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation2_obsProps_Procedure.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation2_obsProps_Procedure_response.xml>

1. An example request and response of the XML implementation of the *GetObservation* operation with a specified observedProperty, procedure and featureOfInterest can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation3_foiIDFilter.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation3_foiIDFilter_response.xml>

1. An example request and response of the XML implementation of the *GetObservation* operation with a specified observedProperty, procedure and spatialFilter can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation4_spatialFilter.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/core/GetObservation4_spatialFilter_response.xml>

## Codes

SOS makes use of code values in several places, for example for identifying the format that observations can be provided in but also which conceptual types for representing features of interest, observations and observation results are supported.

This standard reuses code types defined by the SWE Service Model [OGC 09-001] but also uses additional code types that are defined in the following.

The conceptual model of the *Codes* package is shown in the following UML diagram.

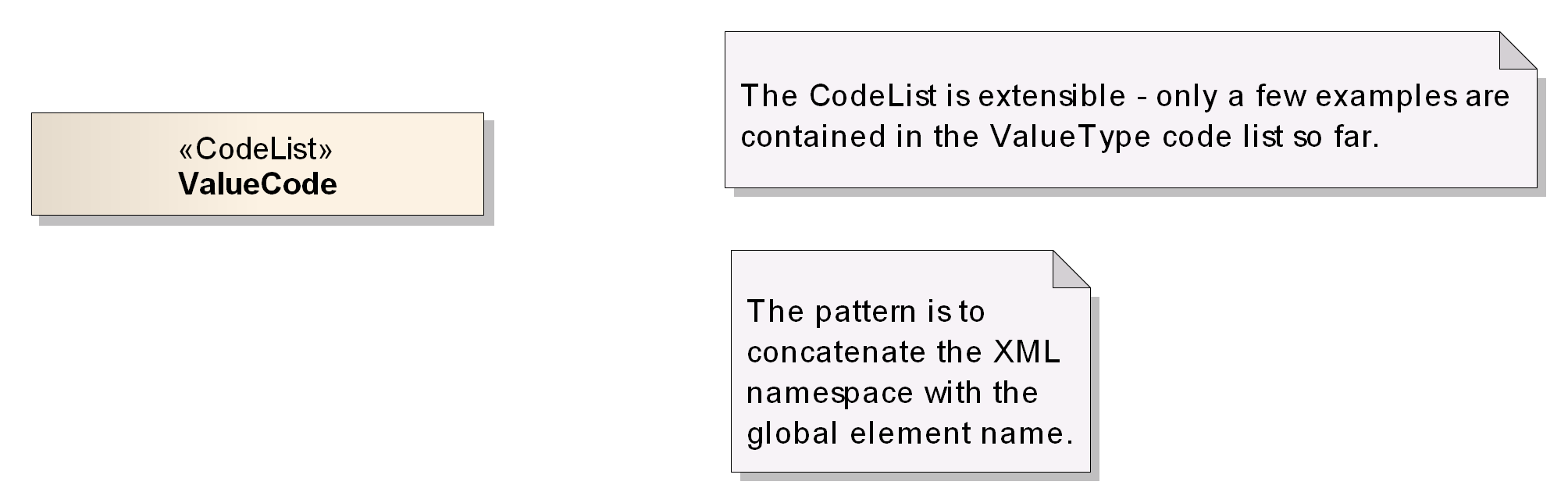


Figure : Data types of Codes package

The ValueCode code list shown in the diagram is empty because it defines an unlimited set of possible code values. The details of each code list contained in the package as well as some example code values will be explained in the following.

### ValueCode

This code list is modeled after the code lists defined in the Common Codes package of the SWE Service Model [OGC 09-001]. As such, all code values assigned to it are URIs.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/core/co-valuecodes** |
| 1. All code values identifying some type defined in a given conceptual model (either directly or via a direct mapping) shall be added to the *ValueCode* code list. These code values shall be URIs. |

There are several ways to identify a type that is defined in a conceptual model:

* The model itself could have assigned a unique identifier to that type. Example: *http://www.opengis.net/def/observationType/OGC-OM/2.0/OM\_TimeSeriesObservation*
* The type can be uniquely identified via a combination of the namespace of the model, the full path to the package that contains the type and the type’s name. Example: *http://www.isotc211.org/19103/2003/BasicTypes/Derived/UnitsOfMeasure/Measure*, *http://www.isotc211.org/19103/2003/BasicTypes/Primitive/Truth/Boolean*
* There is some direct mapping between the conceptual model and some implementation encoding, for example an XML encoding, and the mapped type can be uniquely identified in that encoding. In an XML encoding this could then be the combination of the XML namespace and the element name (or, if that is not available, the complex/simple type name) assigned to the type. Example: *http://www.opengis.net/swe/2.0/Quantity, http://www.opengis.net/gml/3.2/measure*

The following table lists some code values for the *ValueCode* list together with their definition and meaning. Extensions to this specification can define additional codes.

Table – List of some code values used for identifying types defined in a conceptual model

|  |  |
| --- | --- |
| **Value code value** | **Definition/Meaning** |
| http://www.opengis.net/def/observationType/OGC-OM/2.0/OM\_Observation | the basic Observation type defined by O&M  Note: OGC 10-025 table 5 lists URIs to identify other observation types defined in the O&M model |
| http://www.opengis.net/swe/2.0/DataArray | the DataArray type defined by SWE Common [OGC 08-094] |
| http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF\_SamplingPoint | the SF\_SamplingPoint type defined by O&M  Note: OGC 10-025 table 6 lists URIs to identify other sampling feature types defined in the O&M model |

# Enhanced Operations Extension

## Requirements Class: Feature of Interest Retrieval

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |

### GetFeatureOfInterest Operation

The *GetFeatureOfInterest* operation allows clients to retrieve certain features of interest of observations.

The conceptual model of the *GetFeatureOfInterest* operation is shown in the following UML diagram.

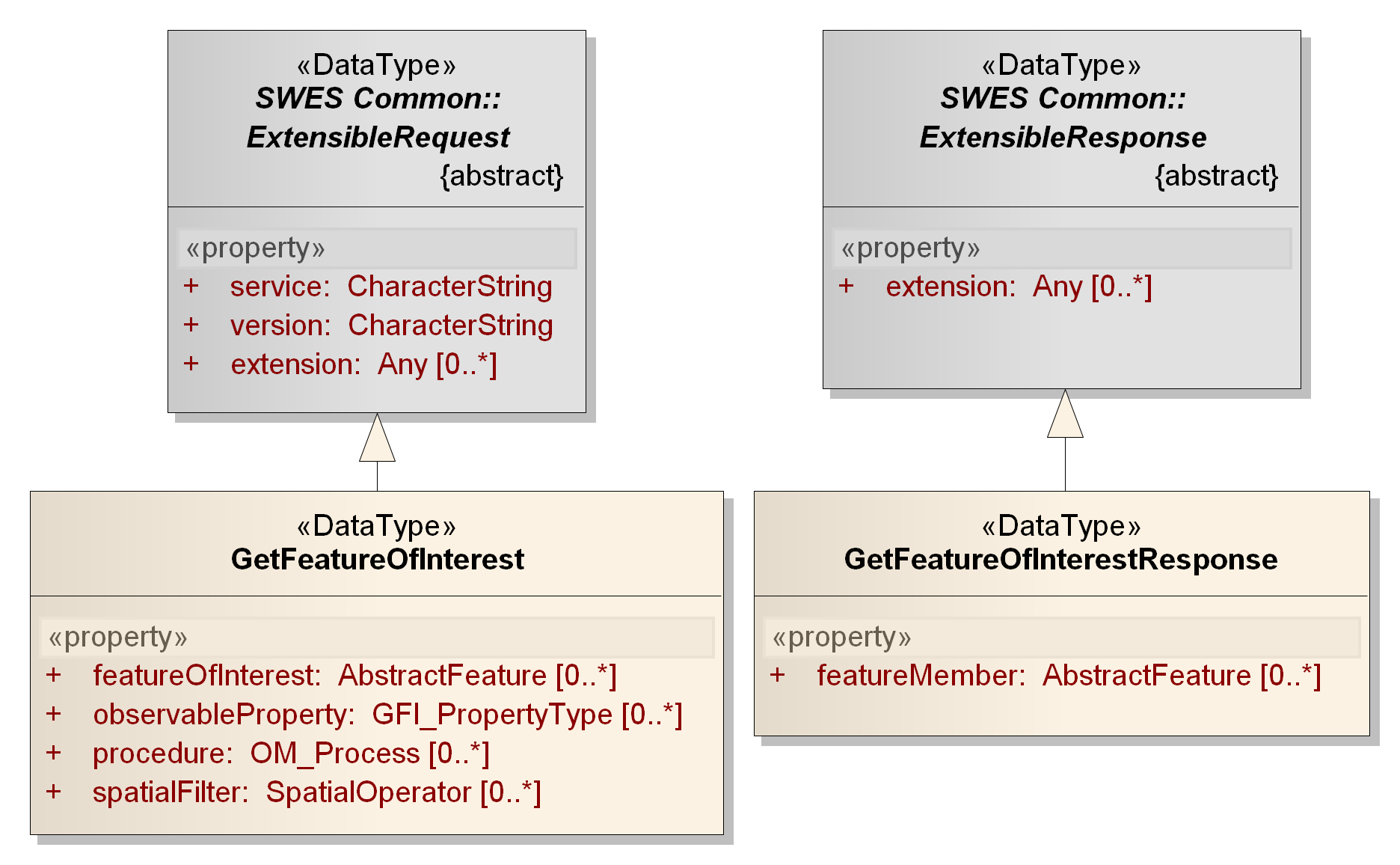


Figure : Data types of GetFeatureOfInterest operation

#### Request

The SOS GetFeatureOfInterest data type derives from the SWES ExtensibleRequest type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/request-structure** |
| In addition to the properties inherited from SWES ExtensibleRequest, the SOS GetFeatureOfInterest operation request type shall include the properties according to   1. Table 21. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/parameters** |
| 1. The SOS returns all features that match the specified parameter values. The request parameters shall be connected with an explicit **AND**. The values of each of the parameters are connected with an implicit **OR**. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/omitting-parameters** |
| 1. If an optional parameter of a *GetFeatureOfInterest* request is not specified by the client, the filter (represented by the parameter) shall not be applied to the feature set which will be returned by the SOS server. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/spatial-filter** |
| 1. The spatialFilter of the request shall be applied to a spatial property of the feature and NOT of the associated observations. The property to which the filter shall be applied is defined in the valueReference element of the spatialFilter. |

NOTE: An implementation of an SOS server may return an exception message as specified in Clause 15 of [OGC 09-001] if the response of a *GetFeatureOfInterest* request would be too big to be reasonably send to a client.

1. Resulting from and an example abstract *GetFeatureOfInterest* request looks like this:

*GetFeatureOfInterest* ( observableProperty := *temperature*

**AND** procedure := *thermometerX* **OR** *anemometerY*)

This request returns all features of interest of all spatial extent, which carry the property “temperature” and are observed by the sensor “thermometerX” or “anemometerY”.

Table : Properties of GetFeatureOfInterest data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| featureOfInterest | Pointer to a feature of interest which shall be returned. | AbstractFeature id  see clause D.3.4 in [ISO 19136] and clause 9.3 in [OGC 07-036] | Zero or many (optional) |
| observableProperty | Pointer to an observable property the requested features shall carry. | GFI\_PropertyType id  see [ISO 19156] | Zero or many (optional) |
| procedure | Pointer to a procedure which is observing the requested features. | OM\_Process id  see [ISO 19156] | Zero or many (optional) |
| spatialFilter | Specifies a filter which applies to a spatial property of the requested features. This property is defined in the valueReference element of the SpatialOperator. | SpatialOperator  see [ISO 19143] | Zero or many (optional) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Response

The SOS GetFeatureOfInterestResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/response-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS GetFeatureOfInterestResponse shall include the property according to. |

Table : Properties of GetFeatureOfInterestResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| featureMember | Feature of interest matching the request parameters specified by the client. | AbstractFeature  (see clause D.3.4 in [ISO 19136] and clause 9.3 in [OGC 07-036]) | Zero or many (optional) |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gfoi/exception** |
| 1. When an SOS server encounters an error while performing a *GetFeatureOfInterest* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetFeatureOfInterest operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

#### Examples

1. An example request of the XML implementation of the *GetFeatureOfInterest* operation with a specified observableProperty and procedure can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/enhancedOperations/GetFOI1.xml>

1. An example request of the XML implementation of the *GetFeatureOfInterest* operation with specified featureOfInterest can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/enhancedOperations/GetFOI2.xml>

1. An example request of the XML implementation of the *GetFeatureOfInterest* operation with a specified observableProperty and spatialFilter can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/enhancedOperations/GetFOI3.xml>

## Requirements Class: Observation Retrieval By ID

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/obsByIdRetrieval** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |

### GetObservationById operation

The *GetObservationByID* operation allows the client to retrieve an observation by passing a pointer to that observation.

The conceptual model of the *GetObservationByID* operation is shown in the following UML diagram.

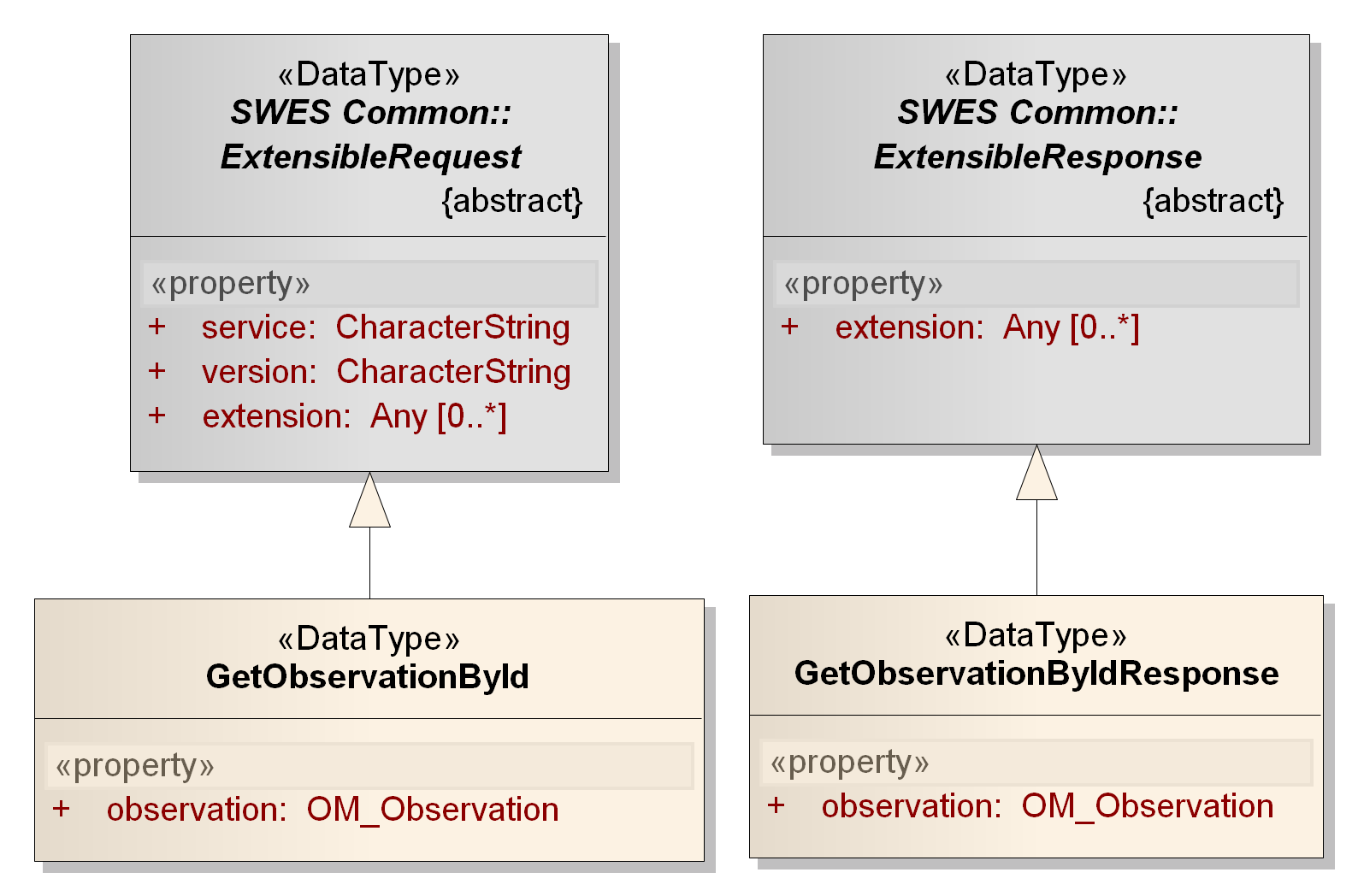


Figure : Data types of GetObservationById operation

#### Request

The SOS GetObservationById data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gobi/request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS GetObservationById shall include the property according to . |

Table : Properties of GetObservationById data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| observation | Pointer to the observation which shall be returned. | OM\_Observation id  see [ISO 19156] | One or many (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Response

The SOS GetObservationByIdResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gobi/response-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS GetObservationByIdResponse shall include the property according to . |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gobi/response-behavior** |
| 1. The GetObservationById operation response shall contain an O&M observation that has a gml:identifier value equal to the value of the observation parameter provided in the operation request. |

Table : Properties of GetObservationByIdResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| observation | O&M Observation matching the observation parameter specified by the client. | OM\_Observation  see [ISO 19156] | One (mandatory) |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gobi/exception** |
| 1. When an SOS server encounters an error while performing a *GetObservationById* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetObservationById operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gobi/exception-no** |
| 1. If no observation matches the client’s request an exception message is returned with the exception code “InvalidParameterValue” and locator value “observation”. |

#### Examples

1. An example request of the XML implementation of the *GetObservationById* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/enhancedOperations/GetObservationById.xml>

# Transactional Extension

## Requirements Class: InsertionCapabilities

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/insertionCap** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |

### InsertionCapabilities Section

The InsertionCapabilities section states which feature types, observation types, and result types are supported by the SOS server for the insertion of new data. It is needed by the *Sensor Insertion* (Subclause ), *Observation Insertion* (Subclause ), and *Result Insertion* (Subclause ) requirements classes.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/insertionCap/structure** |
| 1. The InsertionCapabilities section shall be structured as defined in and . |

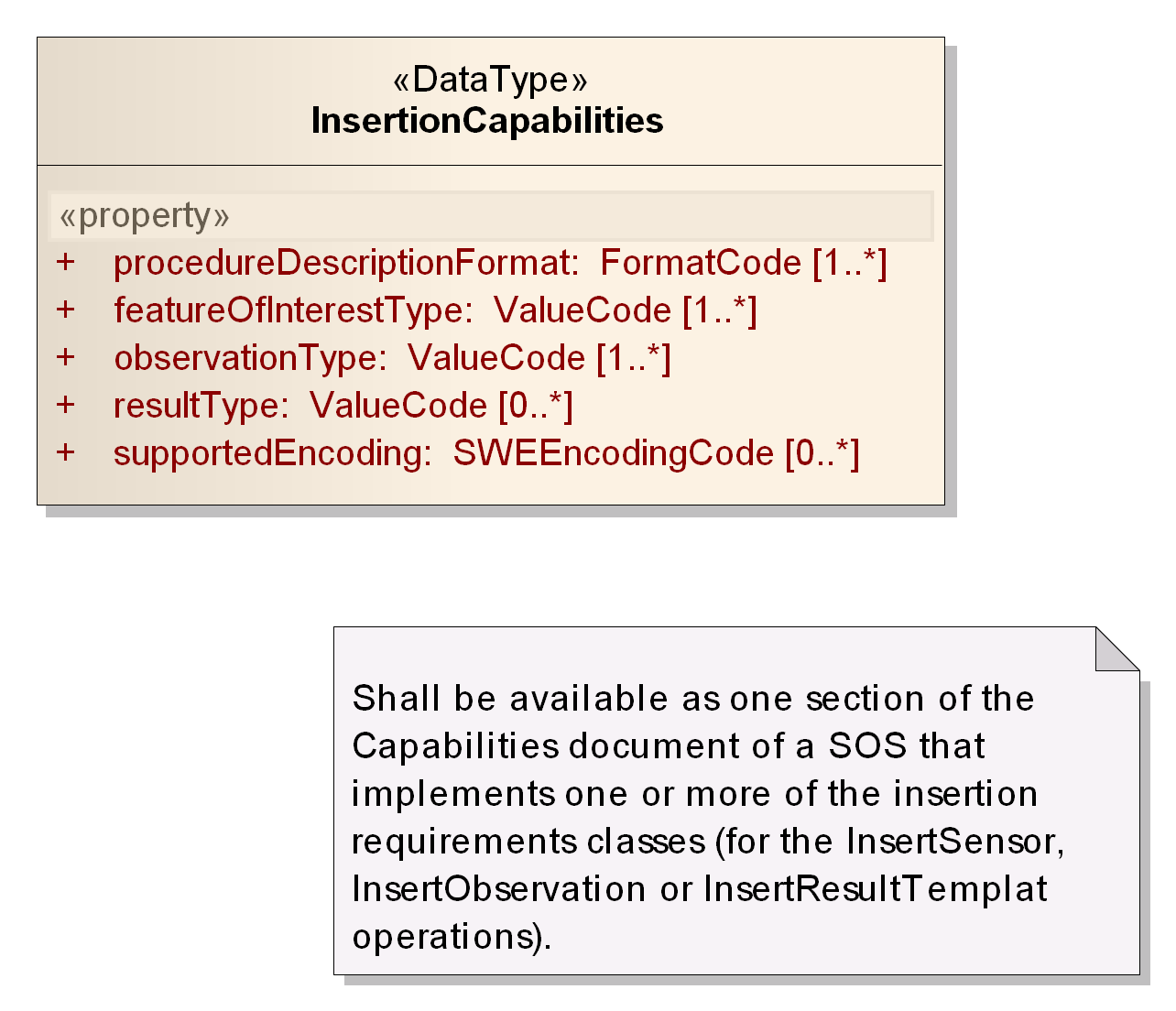


Figure : Data type of InsertionCapabilities section

Table : Properties of InsertionCapabilities data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| procedureDescriptionFormat | identifier of a procedure description format that is supported by the service  Listed procedure descriptions can be used when inserting a new sensor or when inserting an observation (or result template with an observation) that references a procedure encoded in a given description format. | FormatCode  see OGC 09-001 | One or many (optional) |
| featureOfInterestType | identifier of the feature type that supported by the service  Listed feature types can be used when adding a new feature of interest which is associated with an observation inserted through *InsertObservation* or *InsertResultTemplate*. | ValueCode  see clause | One or many (mandatory) |
| observationType | identifier of the observation type supported by the service  Listed observation types can be used when adding new observations through *InsertObservation* or *InsertResultTemplate*. | ValueCode  see clause | One or many (mandatory) |
| resultType | identifier of the result type supported by the service  Listed result types can be used for new results of observations which are added to the SOS. | ValueCode  see clause | Zero or many (optional) |
| supportedEncoding | identifier of the result encoding supported by the service  Listed result encodings can be used for new results of observations which are added to the SOS. | SWEEncodingCode  see OGC 09-001 | Zero or many (optional) |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/insertionCap/capabilities-inclusion** |
| 1. If the service lists the InsertionCapabilities in its OperationsMetadata as a supported value for the sections parameter of the *GetCapabilities* operation and a *GetCapabilities* request includes the sections parameter with this value or the “all” value or the request was made without the sections parameter, then the InsertionCapabilities shall be provided in the Capabilities document that is returned by the service. More specifically, it shall then be placed in the sos:Capabilities/sos:extension property. |

## Requirements Class: Sensor Insertion

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/sensorInsertion** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/insertionCap |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/12 |

### InsertSensor Operation

The *InsertSensor* operation allows a client to register a new sensor system at the SOS. The operation is part of the Sensor Insertion requirements class. Sensor observations can only be inserted for sensors that have first been inserted in the SOS.

The conceptual model of the *InsertSensor* operation is shown in the following UML diagram.

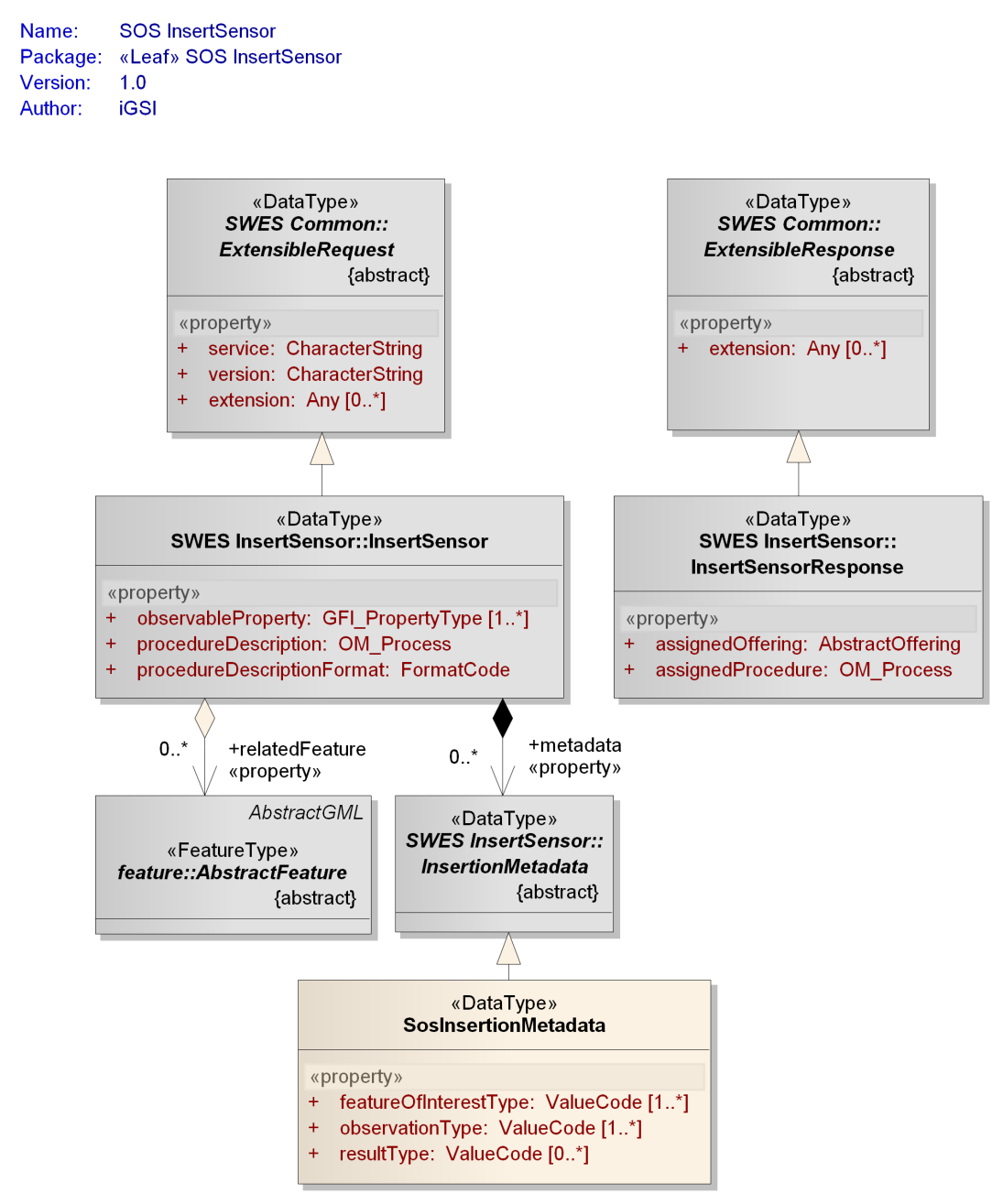


Figure : Data types of InsertSensor operation

The InsertionCapabilities section lists the types which can be used by the client for insertion at the SOS server.

#### Request

The base structure of the *InsertSensor* operation is defined in Clause 13 of [OGC 09-001].

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/is/request-structure** |
| 1. For the SOS the *InsertSensor* operation shall be used as specified in Clause 13 of [OGC 09-001]. Additionally, the SOS extends the abstract InsertionMetadata data type through the type SosInsertionMetadata, with properties as defined in .   This type shall be included in InsertSensor requests sent to the SOS server. Clients shall state in the SosInsertionMetadata element the featureOfInterestTypes, observationTypes and resultTypes which are used by the sensor to encode observations.  The service attribute in the InsertSensor request that is sent to the SOS server shall have the value “SOS” and the version attribute shall have the value “2.0.0”. |

Table Properties of SosInsertionMetadata data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| featureOfInterestType | identifier of feature of interest type associated with observation produced by the sensor | ValueCode  see clause | Zero or many (optional) |
| observationType | identifier of observation type which is produced by the sensor | ValueCode  see clause | One or many (mandatory) |
| resultType | identifier of result type in observations produced by the sensor | ValueCode  see clause | Zero or many (optional)  required if the observation types allow any result type (like the generic observation does) |

#### Response

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/is/response** |
| 1. The *InsertSensor* operation shall return a response as defined in Subclause 13.2.3 of [OGC 09-001]. |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/is/exception** |
| 1. When an SOS server encounters an error while performing an *InsertSensor* operation, it shall return an exception message as specified in Subclause 13.3 of [OGC 09-001]. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/is/exception-unsupported-types** |
| 1. If one of the featureOfInterestType, observationType and resultType specified by the client in the SosInsertionMetadata element provided in the InsertSensor request are not supported by the SOS server (supported values are listed in the InsertionCapabilities section of the Capabilities document) then an exception shall be returned with the exception code “InvalidParameterValue” and locator value “featureOfInterestType”, “observationType” or “resultType”. |

#### Examples

1. An example of request and response of the XML implementation of the *InsertSensor* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/InsertSensor1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/InsertSensor1_response.xml>

## Requirements Class: Sensor Deletion

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/sensorDeletion** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |

### DeleteSensor Operation

The *DeleteSensor* operation allows a client to delete a sensor system from the SOS. The operation is part of the Sensor Deletion requirements class.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ds/request-structure** |
| 1. The *DeleteSensor* request shall be implemented as defined in Clause 14 of the OGC SWE Service Model [OGC 09-001] specification. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ds/obsoffering-deletion** |
| 1. After a successful deletion of a sensor, all ObservationOfferings which have been associated with the deleted sensor shall no longer be listed in the Capabilities of the SOS server. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ds/observation-deletion** |
| 1. When the sensor is successfully deleted by this operation, all observations which have been produced by the sensor shall no longer be accesible or discoverable through the SOS. |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ds/exception** |
| 1. When an SOS server encounters an error while performing a *DeleteSensor* operation, it shall return an exception message as specified in Subclause 14.3 of [OGC 09-001]. |

#### Examples

1. An example of request and response of the XML implementation of the *DeleteSensor* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/DeleteSensor1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/DeleteSensor1_response.xml>

## Requirements Class: Observation Insertion

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/obsInsertion** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/insertionCap |

### InsertObservation Operation

The *InsertObservation* operation allows clients to insert new observations for a registered sensor system.

The conceptual model of the *InsertObservation* operation is shown in the following UML diagram.

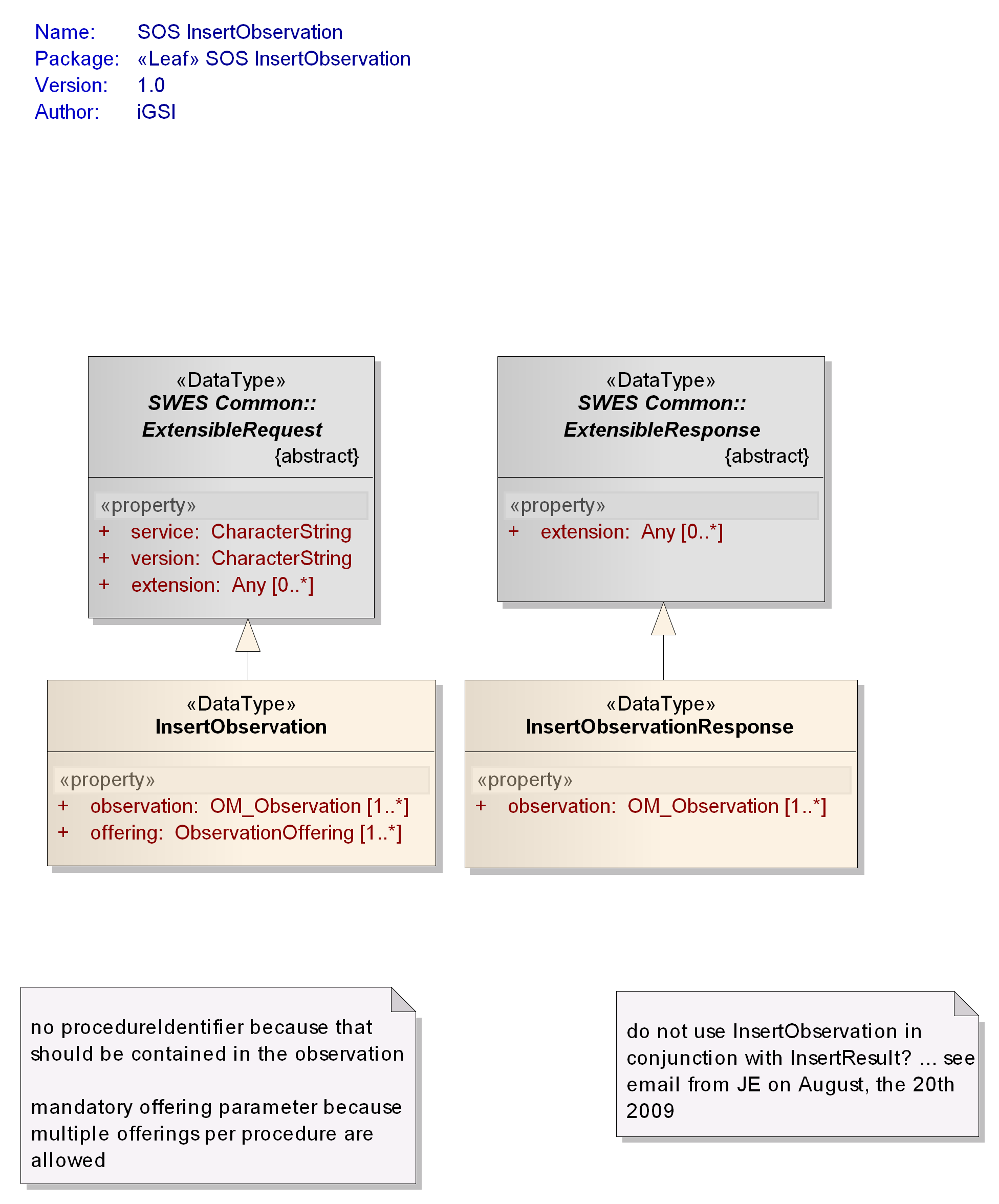


Figure : Data types of InsertObservation operation

The InsertionCapabilities section lists the types which can be used by the client for insertion at the SOS server.

#### Request

The SOS InsertObservation data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS InsertObservation operation request shall include the properties according to . |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/supported-types** |
| 1. The type of the inserted observation and the type of its result shall be supported by the SOS server (and hence listed in the InsertionCapabilities section) AND shall be one of the types defined for (each of) the ObservationOffering(s) to which the observation is added. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/property-constellation** |
| 1. For a particular constellation of procedure, observedProperty and ObservationOffering, observations shall always be encoded in the same observationType and resultType (and also result structure / encoding). The *InsertObservation* operation shall check whether a differing observationType or resultType (or result structure / encoding) is inserted for the same constellation of procedure, observedProperty and ObservationOffering that was inserted before. |

NOTE If this requirement were missing a client could insert observations of type OM\_Observation today and of type OM\_Measurement tomorrow for the same procedure, observedProperty, ObservationOffering constellation. In subsequent calls of *GetObservation* to request data for today and tomorrow it would be unclear how to encode the response.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/multiple-offerings** |
| 1. If multiple offerings are specified for the sensor of the observations which should be inserted, all specified observations shall be added to all specified offerings. |

Table : Properties of InsertObservation data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| offering | Pointer to an ObservationOffering to which the observation shall be added. | ObservationOffering id  see Subclause | One or many (mandatory) |
| observation | Observation to insert | OM\_Observation  see [ISO 19156] | One or many (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Response

The SOS InsertionObservationResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/response-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS InsertionObservationResponse shall include the properties according to . |

Table : Properties of InsertObservationResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| observation | Pointer to the observation which has been inserted. | OM\_Observation id  see [ISO 19156] | One or many (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/exception** |
| 1. When an SOS server encounters an error while performing an *InsertObservation* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the InsertObservation operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/exception-supported-types** |
| 1. If the observationType and resultType of the observation which should be inserted is not supported by the SOS or the types are not listed in the ObservationOfferings of the sensor in the Capabilities, an exception shall be returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, respectively. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/exception-property-constellation** |
| 1. If an observationType or resultType (or result structure / encoding) is inserted for the same constellation of procedure, observedProperty as well as ObservationOffering and that observationType/resultType is different than in previous insertions of observations with that property constellation, an exception shall be returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”. |

#### Examples

1. An example of request and response of the XML implementation of the *InsertObservation* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/InsertObservation1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/transactional/InsertObservation1_response.xml>

# Result Handling Extension

## Requirements Class: Result Insertion

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/resultInsertion** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/insertionCap |

### InsertResultTemplate Operation

The *InsertResultTemplate* operation allows clients to upload a template for result values. Result values which conform to this template can be inserted into the SOS using subsequent calls of the *InsertResult* operation. The *InsertResultTemplate* request includes the pointer to an ObservationOffering into which the results will be inserted. The inserted result template contains not only the description of the result structure and encoding but also an observation template with the complete observation metadata such as procedure, feature of interest and observed property for the observations corresponding to the results.

The conceptual model of the *InsertResultTemplate* operation is shown in the following UML diagram.

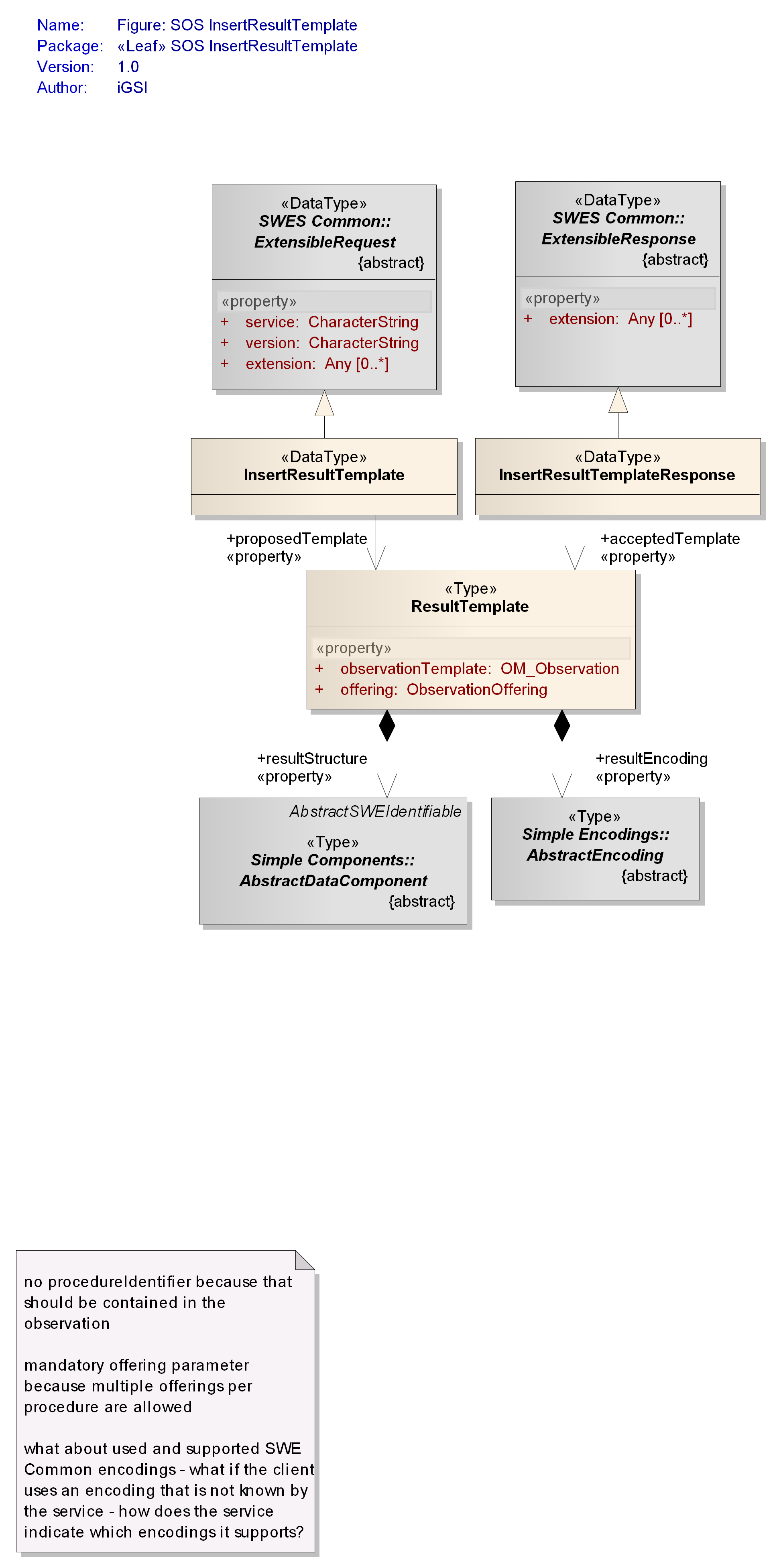


Figure : Data types of InsertResultTemplate operation

The InsertionCapabilities section lists the types which can be used by the client for insertion at the SOS server.

#### Request

The SOS InsertResultTemplate data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS InsertResultTemplate operation request shall include the property according to . |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/supported-types** |
| 1. The type of the registered observation (template), the type of its result and the specified resultEncoding shall be supported by the SOS server (and hence listed in the InsertionCapabilities section) AND shall be one of the types defined for the ObservationOffering to which the observation, built from the results which will be inserted, is added. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/property-constellation** |
| 1. Result values inserted for a particular procedure shall not be encoded in different result encodings, or result structures for the same observedProperty and ObservationOffering. While processing an *InsertResultTemplate* request, an SOS server shall check whether a differing resultStructure or resultEncoding has been used for a constellation of procedure, observedProperty and ObservationOffering that was inserted before. |

NOTE Differing resultEncoding and resultStructure can only be used for the same constellation of observedProperty and procedure if different ObservationOfferings are used.

NOTE If were missing a client could upload today’s results encoded, for example, as a DataArray [OGC 08-094] and tomorrow as a DataRecord [OGC 08-094] for the same procedure, observedProperty, ObservationOffering constellation. In subsequent calls of *GetResult* to request data for today and tomorrow it would be unclear how to encode the response.

Table : Properties of InsertResultTemplate data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| proposedTemplate | Specifies the observation metadata and also information about the structure and encoding of the result, but no result value | ResultTemplate  see | One (mandatory) |

Table : Properties of ResultTemplate data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| offering | Pointer to ObservationOffering to which the results and corresponding observations shall be added. | ObservationOffering id  see Subclause | One (mandatory) |
| observationTemplate | template which contains observation metadata that is used to form complete observation together with result values that are inserted later on | OM\_Observation  see [ISO 19156]  defines further value constraints | One (mandatory) |
| resultStructure | Specifies the structure of the results which will be inserted in subsequent *InsertResult* calls for the observationTemplate | AbstractDataComponent  see [OGC 08-094] | One (mandatory) |
| resultEncoding | Specifies the encoding of the results which will be inserted in subsequent *InsertResult* calls for the observationTemplate | AbstractEncoding  see [OGC 08-094] | One (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/obs-template-structure** |
| 1. The observation that is provided by the client in the ResultTemplate shall have an empty om:phenomenonTime, om:resultTime and om:result. For the first two properties, the nilReason shall be set to the value ‘template’. The procedure, featureOfInterest and observedProperty of the observation template shall not be empty. Other observation properties can be set by the client. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/result-structure-phenomenonTime** |
| 1. The resultStructure in the ResultTemplate shall have at least a swe:Time or swe:TimeRange component with definition property set to the value “http://www.opengis.net/sos/2.0/observation/phenomenonTime”. The value of this component shall be used by the service to populate the om:phenomenonTime property of the observation template for each new result block the client is going to insert via the InsertResult operation. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/result-structure-resultTime** |
| 1. If the resultStructure in the ResultTemplate has a swe:Time or swe:TimeRange component with definition property set to the value “http://www.opengis.net/sos/2.0/observation/resultTime” then the value of this component shall be used by the service to populate the om:resultTime property of the observation template for each new result block the client is going to insert via the InsertResult operation. If no such component is contained in the resultStructure then the service shall use the om:phenomenonTime as value of the om:resultTime (at least the phenomenon time has to be provided in each ResultTemplate). |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/observation-time-provisioning** |
| 1. A client shall encode the om:phenomenonTime as a swe:Time or swe:TimeRange component with definition “http://www.opengis.net/sos/2.0/observation/phenomenonTime”. in the resultStructure that it proposes to the service in the *InsertResultTemplate* operation request. If any of the observation results that the client intends to send to the service via the *InsertResult* operation is going to have a resultTime that is different to the phenomenonTime then the resultStructure of the ResultTemplate shall also have a swe:Time or swe:TimeRange component with definition “http://www.opengis.net/sos/2.0/observation/resultTime”. |

#### Response

The SOS InsertResultTemplateResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/response-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS InsertResultTemplateResponse shall include the property according to . |

Table : Properties of InsertResultTemplateResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Definition | Data type and values | Multiplicity and use |
| acceptedTemplate | Pointer to the ResultTemplate which has been accepted and registered at the SOS server. This ResultTemplate can be used in subsequent *InsertResult* requests. | ResultTemplate id | One (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/exception** |
| 1. When an SOS server encounters an error while performing an *InsertResultTemplate* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the InsertResultTemplate operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/irt/exception-supported-types** |
| 1. If the observationType and resultType of the observation template is not supported by the SOS or the types are not listed in the ObservationOfferings of the sensor in the Capabilities, an exception shall be returned with ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, respectively. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/io/exception-property-constellation** |
| 1. If a differing observationType or resultType (or result structure / encoding) is inserted for the same constellation of procedure, observedProperty and ObservationOffering before,an exception shall be returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”. |

#### Examples

1. An example of request and response of the XML implementation of the *InsertResultTemplate* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/InsertResultTemplate1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/InsertResultTemplate1_response.xml>

### InsertResult Operation

The *InsertResult* operation allows a client to insert new observations for a sensor system by inserting only the results of the observations. The operation is useful if most of the metadata contained in the observations remains the same and/or the communication bandwidth and processing power of the client is limited.

Before invoking the *InsertResult* operation, the sensor has to be associated with the SOS and has to be listed in the Capabilities document. This can be done by invoking the *InsertSensor* operation.

Also, before invoking the *InsertResult* operation the *InsertResultTemplate* operation has to be invoked once for defining the structure of the result elements which are inserted afterwards.

The conceptual model of the *InsertResult* operation is shown in the following UML diagram.

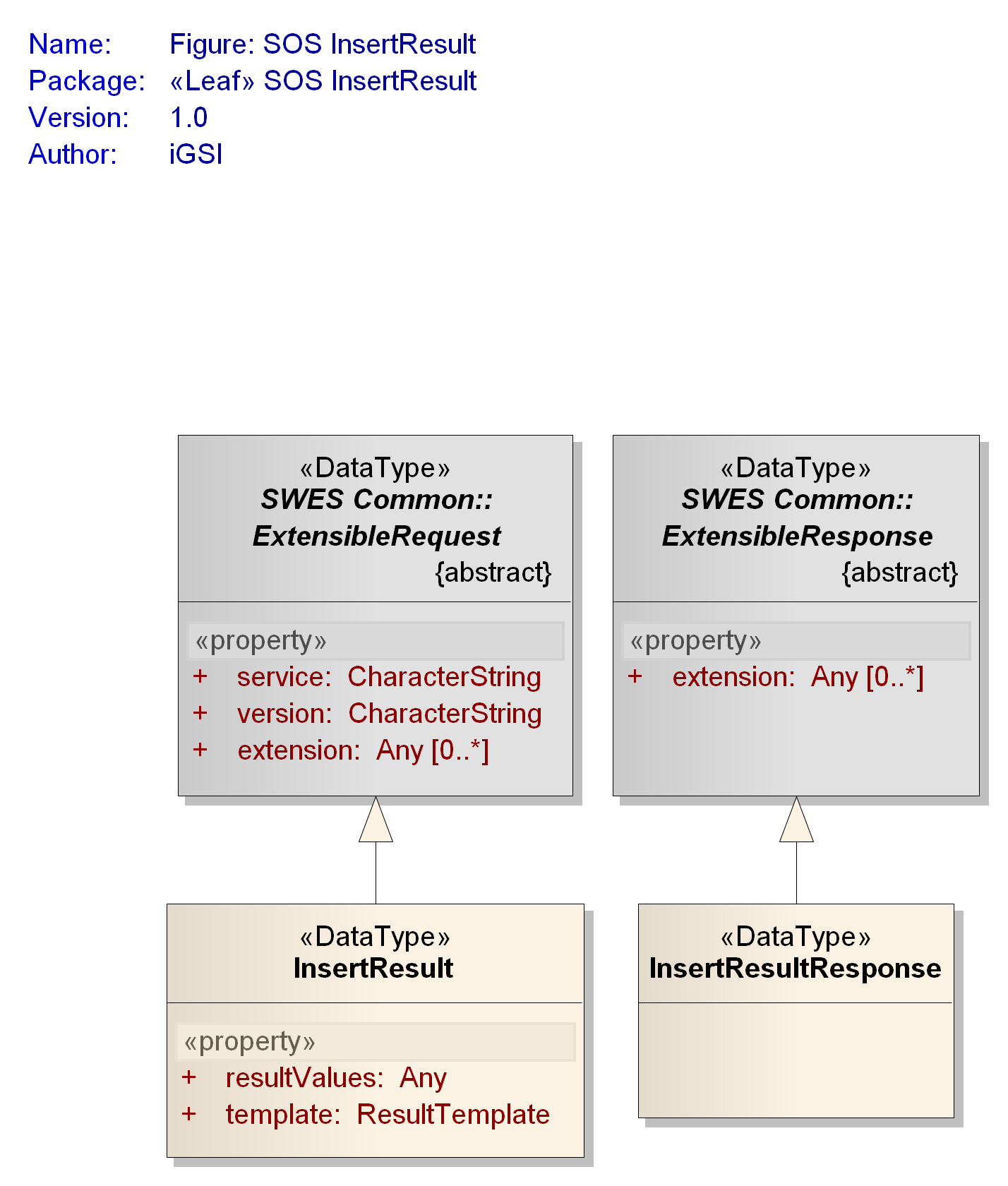


Figure : Data types of InsertResult operation

#### Request

The SOS InsertResult data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ir/request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS InsertResult operation request shall include the properties according to . |

The request includes a pointer to the before uploaded template which describes the structure and encoding of the results. Further, it includes an element which contains the results of observations which shall be inserted.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ir/resultValues-content** |
| 1. The client shall encode the values of the observation result that is to be inserted via the *InsertResult* operation according to the resultStructure of the ResultTemplate it points to in the InsertResult request. The phenomenonTime and resultTime of the observation of which the result is going to be inserted shall be encoded in the result values (as swe:Time or swe:TimeRange component with definition value of “http://www.opengis.net/sos/2.0/observation/phenomenonTime” or “http://www.opengis.net/sos/2.0/observation/resultTime” respectively) so that the service can create a full observation from the information contained in the referenced template and the given result values. |

Table : Properties of InsertResult data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| template | Pointer to the template defining the structure and encoding of the results. | ResultTemplate id  see | One (mandatory) |
| resultValues | The results of observations which shall be inserted. | Any type | One (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Response

The SOS InsertResultResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ir/response** |
| 1. The SOS InsertResultResponse shall contain the properties inherited from SWES ExtensibleResponse. No other properties are defined by this standard for the InsertResultResponse type. The return of an instance of the InsertResultResponse shall indicate a successful insertion of the result values. |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ir/exception** |
| 1. When an SOS server encounters an error while performing an *InsertResult* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the InsertResult operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/ir/exception-unknown-result-structure** |
| 1. If the structure and encoding of the inserted results does not adhere to the resultStructure and resultEncoding defined for the ResultTemplate pointed to in the InsertResult request (and inserted beforehand through an *InsertResultTemplate* call) an exception shall be thrown with the exceptionCode “InvalidParameterValue” with locator value “template”. |

#### Examples

1. An example of request and response of the XML implementation of the *InsertResult* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/InsertResult1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/InsertResult1_response.xml>

## Requirements Class: Result Retrieval

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/resultRetrieval** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | **http://www.opengis.net/spec/SOS/2.0/req/insertionCap** |

### GetResultTemplate Operation

The *GetResultTemplate* operation allows the client to retrieve the structure and encoding of the results which will be returned during later *GetResult* operations for specified observed property and ObservationOffering. By requesting this information from the SOS server, the client is enabled to interpret the result values retrieved from subsequent *GetResult* calls.

Note: there is an inherent assumption that the SWE Common defined result structure of observations belonging to one observation offering is the same if the observed property of these observations is the same. In other words, observations generated by one procedure for a certain observed property may not have a different SWE Common defined result structure unless they are associated with different offerings.

The conceptual model of the *GetResultTemplate* operation is shown in the following UML diagram.

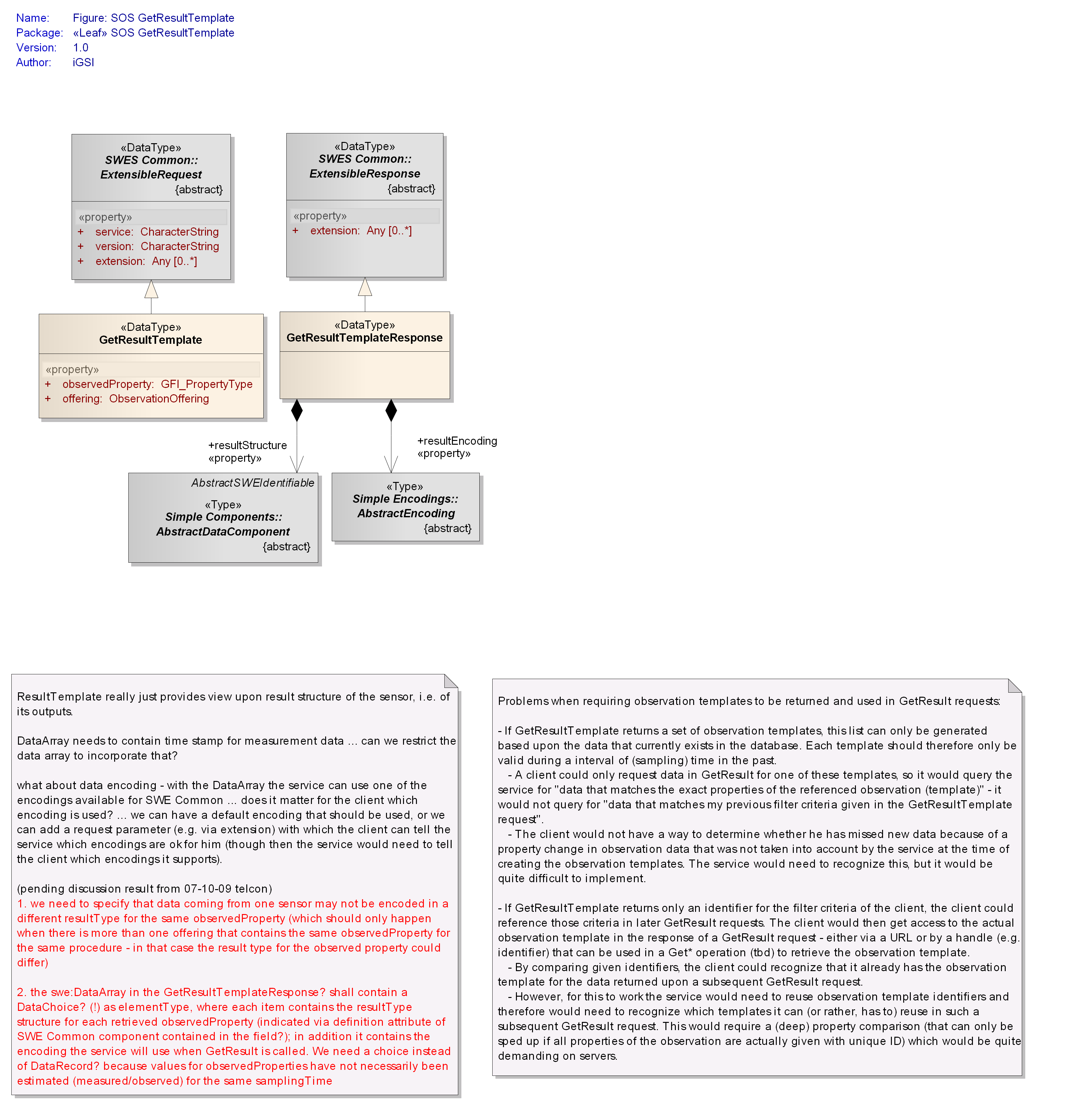


Figure : Data types of GetResultTemplate operation

#### Request

The SOS GetResultTemplate data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/grt/request-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleRequest, the SOS GetResultTemplate operation request shall include the properties according to . |

Table : Properties of GetResultTemplate data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| offering | Pointer to an ObservationOffering for which results will be requested in subsequent *GetResult* calls.  Since an ObservationOffering is associated with exactly one procedure, this parameter indirectly identifies the procedure for which results will be requested. | ObservationOffering id  see Subclause | One (mandatory) |
| observedProperty | Pointer to an observed property for which the results serve values. | GFI\_PropertyType id  see [ISO 19156] | One (mandatory) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details. | | | |

#### Response

The SOS GetResultTemplateResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/grt/response-structure** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS GetResultTemplateResponse shall include the properties according to . |

Table : Properties of GetResultTemplateResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type and values** | **Multiplicity and use** |
| resultStructure | The structure of the results which may be requested in subsequent *GetResult* calls. | AbstractDataComponent  see [OGC 08-094] | One (mandatory) |
| resultEncoding | The encoding of the results which may be requested in subsequent Get*Result* calls. | AbstractEncoding  see [OGC 08-094] | One (mandatory) |

Note: different to the result insertion extension, in the result retrieval extension the intention is not to construct complete observations with the result values returned via the GetResultResponse. Consequently, there is no requirement on SOS services to include components in the sos:GetResultResponse/sos:resultStructure to provide the phenomenon or result time of observations in. Such components may of course be part of an observation result. As such, observations with a SWE Common defined result type can easily be retrieved via the GetResultTemplate/GetResult operations.

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/grt/exception** |
| 1. When an SOS server encounters an error while performing a *GetResultTemplate* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetResultTemplate operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

#### Examples

1. An example of request and response of the XML implementation of the *GetResultTemplate* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/GetResultTemplate1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/GetResultTemplate1_response.xml>

### GetResult Operation

The *GetResult* operation allows retrieving just the result values of observations without the entire metadata of the observation. It offers five parameters: ObservationOffering, feature of interest, observed property, as well as temporal and spatial filter, which can be used to filter the observations of which result values are returned by the SOS. For being able to interpret the returned result values, the client can invoke the *GetResultTemplate*, before calling *GetResult*, to retrieve the structure and encoding of the results returned for the specified ObservationOffering and observed property (the other parameters do not influence the structure or encoding of the results).

The conceptual model of the *GetResult* operation is shown in the following UML diagram.

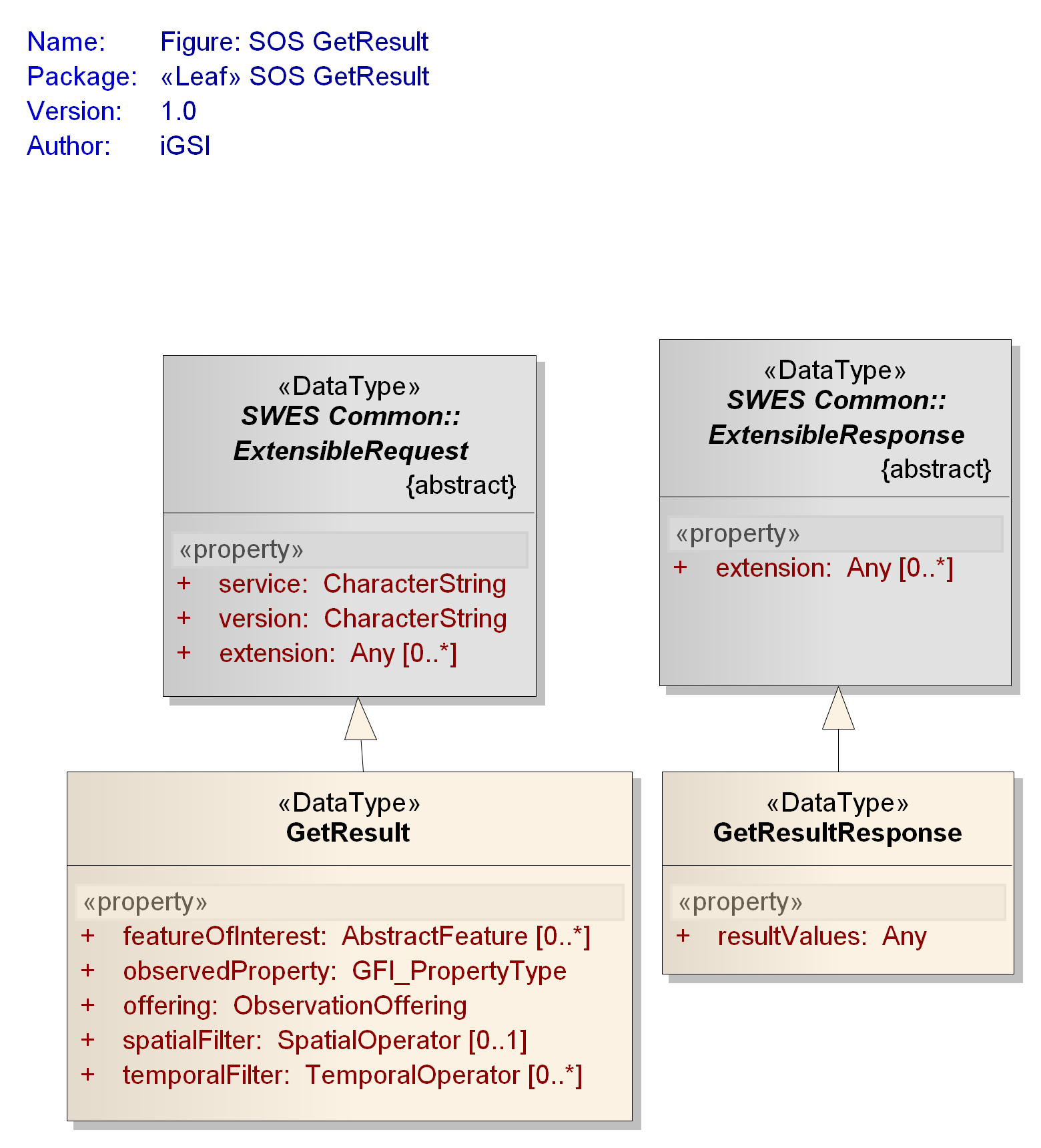


Figure : Data types of GetResult operation

#### Request

The SOS GetResult data type derives from the SWES ExtensibleRequest data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/request-structure** |
| In addition to the properties inherited from SWES ExtensibleRequest, the SOS GetResult operation request shall include the properties according to   1. Table 35. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/parameters** |
| 1. The SOS returns the results of all observations that match the specified parameter values. The request parameters shall be connected with an explicit **AND**. The values of each of the parameters are connected with an implicit **OR**. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/omitting-parameters** |
| 1. If an optional parameter of a *GetResult* request is not specified by the client, the filter (represented by the parameter) shall not be applied to the result set which will be returned by the SOS server. |

NOTE: An implementation of an SOS server may return an exception message as specified in Clause 15 of [OGC 09-001] if the response of a *GetResult* request would be too big to be reasonably send to a client.

1. Resulting from and an example abstract *GetResult* request looks like this:

*GetResult* ( offering := *weatherstation\_in\_my\_backyard*

**AND** observedProperty := *temperature*

**AND** temporalFilter := *January* **OR** *February*)

This request returns the results of the ObservationOffering “weatherstation\_in\_my\_backyard”, of all spatial extent and all features of interest, which were measured for the property “temperature” and were observed in “January” or “February”.

Table : Properties of GetResult data type

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Definition | Data type and values | Multiplicity and use |
| featureOfInterest | Pointer to a feature of interest of the observations whose results are requested. | AbstractFeature id  see clause D.3.4 in [ISO 19136] and clause 9.3 in [OGC 07-036] | Zero or many (optional) |
| offering | Pointer to an ObservationOffering advertised in the Capabilities document for which results are requested. | ObservationOffering id  see | One (mandatory) |
| observedProperty | Pointer to an observedProperty of the observations whose results are requested. | GFI\_PropertyType id  see [ISO 19156] | One (mandatory) |
| temporalFilter | Specifies a filter for a time property of observations whose results are requested.1 | TemporalOperator  see [ISO 19143] | Zero or more (optional) |
| spatialFilter | Specifies a filter2 which applies to a spatial property of an observation (or one of its properties). This property is defined in the valueReference element of the SpatialOperator. | SpatialOperator  see [ISO 19143] | Zero or one (optional) |
| id) Note: the primary use of this property is to provide a pointer/identifier – see OGC 09-001 clause 16.3.1 for further details.  1) The supported time range for the phenomenonTime property of observations is listed in the selected ObservationOffering. The supported temporal operands and operators shall be listed in the FilterCapabilities section of the Capabilities document.  2) A profile of this generic spatialFilter is given in clause . This profile restricts the spatialFilter so that it is applied to the sampling location parameter of the observations. | | | |

#### Response

The SOS GetResultResponse data type derives from the SWES ExtensibleResponse data type defined in OGC 09-001 and inherits its properties.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/response** |
| 1. In addition to the properties inherited from SWES ExtensibleResponse, the SOS GetResultResponse shall include the properties according to |

Table : Properties of GetResultResponse data type

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Definition | Data type and values | Multiplicity and use |
| resultValues | Encoded value blocks representing the result values of the observations targeted by the GetResult request. | Any | One (mandatory) |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/response-empty** |
| 1. If no observations match the parameters of the GetResult request then the resultValues property of the GetResultResponse shall be empty. |

#### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/gr/exception** |
| 1. When an SOS server encounters an error while performing a *GetResult* operation, it shall return an exception message as specified in Clause 15 of [OGC 09-001] with applicable exception code as defined in . The meaning of each exception code shall be as defined in OGC 06-121r3 and OGC 09-001. |

Listing : exception codes applicable to the GetResult operation

* OperationNotSupported
* MissingParameterValue
* InvalidParameterValue
* OptionNotSupported
* NoApplicableCode
* InvalidRequest
* RequestExtensionNotSupported

#### Examples

1. An example of request and response of the XML implementation of the *GetResult* operation can be found here:

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/GetResult1.xml>

<http://test.schemas.opengis.net/sos/2.0/examples/resultHandling/GetResult1_response.xml>

# Spatial Filtering Profile

In this profile of the SOS 2.0, the observations offered by an SOS server are restricted to spatial observations which provide a well-defined parameter for carrying the sampling geometry of an observation. If an SOS server follows this profile, spatial filters can target the sampling geometries of observations.

## Requirements Class: Spatial Filtering Profile

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/OMXML/2.0/req/SpatialObservation |

The requirements class “Spatial Observation Data” (http://www.opengis.net/spec/OMXML/2.0/req/SpatialObservation) defined in [OGC 10-025] Subclause 7.13 constrains the encoding of an observation to have one parameter which contains the sampling geometry of the observation. The sampling geometry represents the spatial extent where the observation result applies to. This is usually the extent of the observation's feature of interest but may also be computed or determined by other means.

1. An example of such a spatial observation as defined by [OGC 10-025] with sampling location parameter is shown here:

<?xml version="1.0" encoding="UTF-8"?>

<om:OM\_Observation>

…

<om:parameter>

<om:NamedValue>

<om:name

xlink:href="http://www.opengis.net/req/omxml/2.0/data/samplingGeometry"/>

<om:value>

<gml:Point gml:id="SamplingPoint">

<gml:pos srsName="urn:ogc:def:crs:EPSG::4326">52.9 7.52</gml:pos>

</gml:Point>

</om:value>

</om:NamedValue>

</om:parameter>

…

</om:OM\_Observation>

The complete example can be found at:

<https://svn.opengeospatial.org/ogc-projects/sp/swe.umbrella.swg/xsd/om/2.0/examples/spatial_observation1.xml>

The following requirement restricts the observations served by an SOS server to those spatial observations. The sampling geometry of an observation is encoded as a parameter of the observation.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile/observationRestriction** |
| 1. All observations provided by an SOS server conformant to this extension shall be encoded compliant to the conformance class   http://www.opengis.net/spec/OMXML/2.0/conf/SpatialObservation as defined in [OGC 10-025] Subclause 7.13. |

The observedArea provided in the Capabilities is restricted to represent the boundary of the sampling geometries of the observations.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile/observedAreaRestriction** |
| The observedArea of ObservationOfferings provided in the Contents section of the Capabilities document shall contain a geometry representing the boundary of the sampling geometries of the observations that belong to the offering. |

The spatial filter of a *GetObservation* or, if supported, of a *GetResult* operation request can target the sampling geometries of the observations by choosing the following value for the valueReference of the spatialFilter:

om:parameter/om:NamedValue[om:name/@xlink:href= 'http://www.opengis.net/req/omxml/2.0/data/samplingGeometry’]/om:value

1. An example of request of the XML implementation of the *GetObservation* operation conformant to this profile can be found at:

<http://test.schemas.opengis.net/sos/2.0/examples/spatialFilteringProfile/GetObservation1_spatialFilteringProfile.xml>

# Binding Extension

This Clause specifies concrete encodings and bindings for the operations of the SOS 2.0 standard. Those bindings describe how SOS 2.0 clients and servers can communicate with each other.

## Requirements Class: XML Encoding

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/xml** | |
| **Target Type** | XML Instances |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/SWE/2.0/req/xsd-simple-components |
| **Dependency** | http://www.opengis.net/spec/SWE/2.0/req/xsd-simple-encodings |
| **Dependency** | http://www.opengis.net/spec/OMXML/2.0/req/observation |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/AnnexB |
| **Dependency** | http://www.opengis.net/doc/IS/OWS/1.1/clause/AnnexB |
| **Dependency** | urn:iso:ts:iso:19143:clause:AnnexB |
| **Dependency** | urn:iso:ts:iso:19136:clause:AnnexC |

In addition to this document, this standard includes several normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. After OGC acceptance of this standard, these XML Schema Documents will also be posted online at the URL http://schemas.opengis.net/sos/2.0.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative.

The UML model has been mapped to its XML Schema encoding using the rules described in OGC 09-001, resulting in the following XML Schema documents:

sos.xsd (includes the other schema through xs:include statements)

sosContents.xsd

sosGetCapabilities.xsd

sosGetFeatureOfInterest.xsd

sosGetObservation.xsd

sosGetObservationById.xsd

sosGetResult.xsd

sosGetResultTemplate.xsd

sosInsertionCapabilities.xsd

sosInsertObservation.xsd

sosInsertResult.xsd

sosInsertResultTemplate.xsd

sosInsertSensor.xsd

The following table provides an overview how each of the conceptual model types defined by this standard has been realized in the XML Schema implementation.

Note: as the types defined in the *SOS Codes* package are not intended to be encoded as XML elements, an XML Schema file for that package is not needed and thus not available.

Table : XML schema implementation of types defined by this standard

| **UML class** | **object element** | **type** | **property type** |
| --- | --- | --- | --- |
| *SOS GetCapabilities* | | | |
| Capabilities | sos:Capabilities | sos:CapabilitiesType | sos:CapabilitiesPropertyType |
| Contents | sos:Contents | sos:ContentsType | sos:ContentsPropertyType |
| GetCapabilities | sos:GetCapabilities | sos:GetCapabilitiesType | sos:GetCapabilitiesPropertyType |
| ObservationOffering | sos:ObservationOffering | sos:ObservationOfferingType | sos:ObservationOfferingPropertyType |
| *SOS GetFeatureOfInterest* | | | |
| GetFeatureOfInterest | sos:GetFeatureOfInterest | sos:GetFeatureOfInterestType | sos:GetFeatureOfInterestPropertyType |
| GetFeatureOfInterestResponse | sos:GetFeatureOfInterestResponse | sos:GetFeatureOfInterestResponseType | sos:GetFeatureOfInterestResponsePropertyType |
| *SOS GetObservation* | | | |
| GetObservation | sos:GetObservation | sos:GetObservationType | sos:GetObservationPropertyType |
| GetObservationResponse | sos:GetObservationResponse | sos:GetObservationResponseType | sos:GetObservationResponsePropertyType |
| *SOS GetObservationById* | | | |
| GetObservationById | sos:GetObservationById | sos:GetObservationByIdType | sos:GetObservationByIdPropertyType |
| GetObservationByIdResponse | sos:GetObservationByIdResponse | sos:GetObservationByIdResponseType | sos:GetObservationByIdResponsePropertyType |
| *SOS GetResult* | | | |
| GetResult | sos:GetResult | sos:GetResultType | sos:GetResultPropertyType |
| GetResultResponse | sos:GetResultResponse | sos:GetResultResponseType | sos:GetResultResponsePropertyType |
| *SOS GetResultTemplate* | | | |
| GetResultTemplate | sos:GetResultTemplate | sos:GetResultTemplateType | sos:GetResultTemplatePropertyType |
| GetResultTemplateResponse | sos:GetResultTemplateResponse | sos:GetResultTemplateResponseType | sos:GetResultTemplateResponsePropertyType |
| *SOS InsertObservation* | | | |
| InsertObservation | sos:InsertObservation | sos:InsertObservationType | sos:InsertObservationPropertyType |
| InsertObservationResponse | sos:InsertObservationResponse | sos:InsertObservationResponseType | sos:InsertObservationResponsePropertyType |
| *SOS InsertResult* | | | |
| InsertResult | sos:InsertResult | sos:InsertResultType | sos:InsertResultPropertyType |
| InsertResultResponse | sos:InsertResultResponse | sos:InsertResultResponseType | sos:InsertResultResponsePropertyType |
| *SOS InsertResultTemplate* | | | |
| InsertResultTemplate | sos:InsertResultTemplate | sos:InsertResultTemplateType | sos:InsertResultTemplatePropertyType |
| InsertResultTemplateResponse | sos:InsertResultTemplateResponse | sos:InsertResultTemplateResponseType | sos:InsertResultTemplateResponsePropertyType |
| ResultTemplate | sos:ResultTemplate | sos:ResultTemplateType | sos:ResultTemplatePropertyType |
| *SOS InsertSensor* | | | |
| SosInsertionMetadata | sos:SosInsertionMetadata | sos:SosInsertionMetadataType | sos:SosInsertionMetadataPropertyType |
| *SOS InsertionCapabilities* | | | |
| InsertionCapabilities | sos:InsertionCapabilities | sos:InsertionCapabilitiesType | sos:InsertionCapabilitiesPropertyType |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/xml/GeneralEncodingRules** |
| 1. The XML encoding of the conceptual types defined in this standard shall be as defined by the XML Schema files listed and referenced in subclause . More specifically, the XML encoding of each conceptual type shall be valid against the XML Schema definition of the according mapping as defined in . |

## Requirements Class: Core KVP Binding

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | urn:ietf:2396 |
| **Dependency** | urn:iso:ts:iso:8601 |
| **Dependency** | http://www.opengis.net/doc/IS/OWS/1.1/clause/10.2.3 |

This requirements class defines how to invoke SOS 2.0 operations over HTTP GET with key/value pair (KVP) encoding. This KVP binding is defined for the operations *GetCapabilities*, *DescribeSensor*, and *GetObservation*.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/general** |
| 1. For this KVP binding, the general rules defined in Subclause 11.5 of [OGC 06-121r3] apply if not superseded by the here defined requirements. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/url-encoded-values** |
| 1. Special characters of parameter values of the KVP request shallbeURL-encoded as defined in [IETF 2396]. |

1. Use “%20” to represent a whitespace, “ ”.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/case-sensitivity** |
| 1. Keys shallbe case insensitive, values shallbe case sensitive. |

Any extension property contained in the conceptual model of SOS operation request and response types can be realized as simple additional key-value pairs that are appended to the parameters of the original operation. Such extension kvp parameters can be defined in extensions to this requirements class.

### GetCapabilities KVP Binding

This KVP binding is a realization of the conceptual model of the *GetCapabilities* operation as defined in Subclause . In general, all requirements defined for the conceptual model of the *GetCapabilities* operation apply for its realization as a KVP binding.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/gc-request** |
| 1. The KVP encoding of the *GetCapabilities* operation request shallbe as specified in Table 5 in Subclause 7.2.3 of [OGC 06-121r3]. The fixed value of the service parameter shall be “SOS”. |

1. To request a Capabilities document, a client can issue the following minimal *GetCapabilities* operation request encoded as KVP:

http://hostname:port/path?service=SOS&request=GetCapabilities&AcceptVersions=2.0.0

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/gc-response** |
| 1. The response behaviour of the *GetCapabilities* operation shallbe implemented as defined in Subclause of this document and encoded as defined in Subclause . |

### DescribeSensor KVP Binding

This KVP binding is a realization of the conceptual model of the *DescribeSensor* operation as defined in Subclause . In general, all requirements defined for the conceptual model of the *DescribeSensor* operation apply for its realization as a KVP binding. However, due to the limitations of a KVP binding, the parameterization of the operation needs to be restricted and further defined as described below.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-request** |
| 1. The KVP encoding of a *DescribeSensor* request shallbe as defined in . |

Table : DescribeSensor request KVP encoding

|  |  |  |
| --- | --- | --- |
| **Name** | **Definition and format** | **Optionality** |
| service | Identifier of the OGC service.  Fixed value: “SOS”. | Mandatory |
| version | Request protocol version.  Fixed value “2.0.0”. | Mandatory |
| request | Request type name. Fixed value: “DescribeSensor”. | Mandatory |
| procedure | URL-encoded URI pointing to the procedure for which a metadata description shall be retrieved. | Mandatory |
| procedureDescriptionFormat | URL-encoded URI pointing to the requested procedure description format.  Recommended by this standard is SensorML  SensorML version 1.0.1 [OGC 07-022r2] is identified by the value “http%3A%2F%2Fwww.opengis.net%2FsensorML%2F1.0.1” (the URL encoded namespace) | Mandatory |
| validTime | Time instance or time interval encoded conformant to [ISO 8601] for which the sensor description shall be retrieved.  If omitted the currently valid sensor description shall be returned.  If end time of time interval is in the future, all descriptions from start time to now - including the current description - shall be returned | Optional |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-time-encoding** |
| 1. The validTime request parameter shall be encoded compliant to [ISO 8601]. |

1. The following KVP structure requests metadata about a procedure:

http://www.myserver.org:port/path

?service=SOS

&version=2.0.0

&request=DescribeSensor

&procedure=urn:ogc:object:Sensor:MyOrg:thermometer1

&procedureDescriptionFormat=http%3A%2F%2Fwww.opengis.net%2FsensorML%2F1.0.1

&validTime=2010-01-01T18:31:42

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-response** |
| 1. The response behaviour of the *DescribeSensor* operation shallbe as defined in Subclause of this document. The operation response shall be encoded as defined in Annex B of OGC 09-001 |

### GetObservation KVP Binding

This KVP binding is a realization of the conceptual model of the *GetObservation* operation as defined in Subclause . In general, all requirements defined for the conceptual model of the *GetObservation* operation apply for its realization as a KVP binding. However, due to the limitations of a KVP binding, the parameterization of the operation needs to be restricted and further defined as described below.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-request** |
| 1. The KVP encoding of a *GetObservation* request shallbe as defined in Table 39. |

Table : GetObservation request KVP encoding

|  |  |  |
| --- | --- | --- |
| **Name** | **Definition and format** | **Optionality** |
| service | Identifier of the OGC service.  Fixed value: “SOS”. | Mandatory |
| version | Request protocol version.  Fixed value “2.0.0”. | Mandatory |
| request | Request type name.  Fixed value: “GetObservation”. | Mandatory |
| offering | A URL-encoded URI pointing to the requested ObservationOffering. | Mandatory |
| observedProperty | Comma-separated unordered list of one or more URL-encoded URIs pointing to the observed properties of the requested observations. | Optional |
| procedure | Comma-separated unordered list of one or more URL-encoded URIs pointing to procedures of the requested observations. | Optional |
| featureOfInterest | Comma-separated unordered list of one or more URL-encoded URIs pointing to features of interest of the requested features of interest. | Optional |
| spatialFilter | Specifies a bounding box used as a spatial filter which applies to a spatial property (identified by its valueReference) of an observation.  The bounding box shall be encoded as defined in . | Optional |
| temporalFilter | Specifies a temporal filter for the phenomenonTime property of requested observations.  The value shall be encoded compliant to [ISO 8601]. Instance and Periods are supported. Periods of time (start and end) are separated by “/”. | Optional |
| responseFormat | Specifies the desired response format for transport of the observations. The supported output formats are listed in the ObservationOffering in the Capabilities document.  By default this is O&M 2.0 [OGC 10-025] identified by the value “http%3A%2F%2Fwww.opengis.net%2Fom%2F2.0” (URL-encoded namespace). | Optional |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-bbox-encoding** |
| 1. For the KVP binding, a bounding box shall be used as the spatialFilter as defined in the conceptual model of the *GetObservation* operation (Subclause ).   The encoding of the bounding box shall be a list of comma separated values. The first value shall be the valueReference of the spatial property of the observations to which this bounding box, as a spatial filter, is applied.  The following values shall be as defined in in Subclause 10.2.3 of [OGC 06-121r3].  This results in the following encoding:  valueReference,minCoordinate1,minCoordinate2,...,minCoordinateN,maxCoordinate1,maxCoordinate2,...,maxCoordinateN,crsURI  The crsURI is optional. An example value for crsURI is “urn:ogc:def:crs:EPSG::4326”. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-temporalFilter-encoding** |
| 1. The temporalFilter parameter shall be used to filter on the phenomenonTime property of requested observations. The value of temporalFilter parameter shall be encoded compliant to [ISO 8601]. Instances and periods of time shall be supported. |

1. The following example shows a KVP-encoded *GetObservation* request:

http://myserver.org:port/path

?service=SOS

&version=2.0.0

&request=GetObservation

&offering=thermometer1\_observations

&observedProperty=urn:ogc:def:property:OGC:AirTemperature

&procedure=urn:ogc:object:Sensor:MyOrg:thermometer1

&featureOfInterest=urn:ogc:object:Feature:MyOrg:buildingX

&spatialFilter=om:featureOfInterest/sams:shape,22.32,11.2,32.32,22.2,urn:ogc:def:crs:EPSG::4326

&temporalFilter=2009-01-10T10:00:00Z/2009-01-10T11:00:00Z

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-response** |
| 1. The response behaviour of the *GetObservation* operation shallbe as defined in Subclause of this document. If the response format chosen in the request requires that a GetObservationResponse be returned then it shall be encoded as defined in subclause |

## Requirements Class: GetFeatureOfInterest KVP Binding

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/kvp-core |

This requirements class defines a KVP binding for the *GetFeatureOfInterest* operation. The KVP binding depends on the requirements class “http://www.opengis.net/spec/SOS/2.0/req/kvp-core” and hence inherits the general requirements defined under Subclause .

The here defined KVP binding of the *GetFeatureOfInterest* operation is a realization of the conceptual model of that operation as defined in Subclause . In general, all requirements defined for the conceptual model of the *GetFeatureOfInterest* operation apply for its realization as a KVP binding. However, due to the limitations of a KVP binding, the parameterization of the operation needs to be restricted and further defined as described below.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi/request** |
| 1. The KVP encoding of a *GetObservation* request shallbe as defined in . |

Table : GetFeatureOfInterest request KVP encoding

|  |  |  |
| --- | --- | --- |
| **Name** | **Definition and format** | **Optionality** |
| service | Identifier of the OGC service.  Fixed value: “SOS”. | Mandatory |
| version | Request protocol version.  Fixed value “2.0.0”. | Mandatory |
| request | Request type name.  Fixed value: “GetFeatureOfInterest”. | Mandatory |
| featureOfInterest | Comma-separated unordered list of one or more URL-encoded URIs pointing to requested features of interest. | Optional |
| observableProperty | Comma-separated unordered list of one or more URL-encoded URIs pointing to observable properties which are properties of the requested features of interest. | Optional |
| procedure | Comma-separated unordered list of one or more URL-encoded URIs pointing to procedures which observe the requested features of interest. | Optional |
| spatialFilter | Specifies a bounding box used as a spatial filter which applies to a spatial property (identified by its valueReference) of the requested features.  The bounding box shall be encoded as defined in . | Optional |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi/bbox-encoding** |
| 1. For the KVP binding, a bounding box shall be used as the spatialFilter as defined in the conceptual model of the *GetFeatureOfInterest* operation (Subclause ).   The encoding of the bounding box shall be a list of comma separated values. The first value shall be the valueReference of the spatial property of the features to which this bounding box, as a spatial filter, is applied.  The following values shall be as defined in in Subclause 10.2.3 of [OGC 06-121r3].  This results in the following encoding:  valueReference,minCoordinate1,minCoordinate2,...,minCoordinateN,maxCoordinate1,maxCoordinate2,...,maxCoordinateN,crsURI  The crsURI is optional. An example value for crsURI is “urn:ogc:def:crs:EPSG::4326”. |

1. The following example shows a KVP-encoded *GetFeatureOfInterest* request:

http://myserver.org:port/path

?service=SOS

&version=2.0.0

&request=GetFeatureOfInterest

&observableProperty=urn:ogc:def:property:OGC:AirTemperature

&procedure=urn:ogc:object:Sensor:MyOrg:thermometer1

&spatialFilter=sams:shape,189000,834000,285000,962000,urn:ogc:def:crs:OGC:1.3:CRS84

## Requirements Class: SOAP Binding

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/SOS/2.0/req/soap** | |
| **Target Type** | Web Application |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/core |
| **Dependency** | http://www.opengis.net/spec/SOS/2.0/req/xml |
| **Dependency** | http://www.opengis.net/doc/IS/SWES/2.0/clause/18 |

This requirements class defines the realization of functionality defined by the SOS 2.0 standard for a service using SOAP. This standard does not prescribe usage of either SOAP 1.1 or SOAP 1.2. It also does not prescribe WSDL 1.1 or WSDL 2.0.

This standard does not define any specific policy statements to be included in a WSDL document or in service requests and responses for defining certain established, available or desired behavior. If the need for such policies arises in the future, necessary policy statements can be included in the standard.

### Exceptions

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/soap/exceptions** |
| 1. The operations defined in this standard shall use the exception codes defined by Clause 8 in [OGC 06-121r3] and Clause 15 in [OGC 09-001]. The encoding of these exceptions for the operations defined by this standard (in a SOAP binding) shall be as defined in clause 19.2 of [OGC 09-001]. |

### Action URIs

For the SOAP binding, a standard needs to define action URIs for the following features:

* as SOAPAction HTTP header field of a SOAP 1.1 request
* as action parameter in a SOAP 1.2 request (SOAP 1.2 feature: “http://www.w3.org/2003/05/soap/features/action/”)
* as WS-Addressing [action] message addressing property

NOTE If and how a service instance makes use of one or more of these features depends upon the chosen SOAP and WSDL version as well as on the requirements of the service instance.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/soap/action-uris** |
| 1. Action URIs for the message facets (requests and responses of operations) defined by this standard shall be as defined by in this document. |

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/soap/action-uris-exceptions** |
| 1. Action URIs for exceptions / fault message types, of which SOS operations make use of, shall be as defined in Table 36 (Subclause 19.3) of [OGC 09-001]. |

Table : Action URIs for SOS message facets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Message  Facet a** | **Action URI a** | **Applicable in feature (Y=yes, N=no)** | | |
| **SOAP 1.1 SOAPAction** | **SOAP 1.2 action** | **WS-Addressing [action]** |
| GetCapabilities request | http://www.opengis.net/sos/2.0/GetCapabilities | Y | Y | Y |
| GetCapabilities response | http://www.opengis.net/sos/2.0/GetCapabilitiesResponse | N | N | Y |
| GetObservation request | http://www.opengis.net/sos/2.0/GetObservation | Y | Y | Y |
| GetObservation response | http://www.opengis.net/sos/2.0/GetObservationResponse | N | N | Y |
| GetFeatureOfInterest request | http://www.opengis.net/sos/2.0/GetFeatureOfInterest | Y | Y | Y |
| GetFeatureOfInterest response | http://www.opengis.net/sos/2.0/GetFeatureOfInterestResponse | N | N | Y |
| GetObservationById request | http://www.opengis.net/sos/2.0/GetObservationById | Y | Y | Y |
| GetObservationById response | http://www.opengis.net/sos/2.0/GetObservationByIdResponse | N | N | Y |
| InsertObservation request | http://www.opengis.net/sos/2.0/InsertObservation | Y | Y | Y |
| InsertObservation response | http://www.opengis.net/sos/2.0/InsertObservationResponse | N | N | Y |
| InsertResultTemplate request | http://www.opengis.net/sos/2.0/InsertResultTemplate | Y | Y | Y |
| InsertResultTemplate response | http://www.opengis.net/sos/2.0/InsertResultTemplateResponse | N | N | Y |
| InsertResult request | http://www.opengis.net/sos/2.0/InsertResult | Y | Y | Y |
| InsertResult response | http://www.opengis.net/sos/2.0/InsertResultResponse | N | N | Y |
| GetResultTemplate request | http://www.opengis.net/sos/2.0/GetResultTemplate | Y | Y | Y |
| GetResultTemplate response | http://www.opengis.net/sos/2.0/GetResultTemplateResponse | N | N | Y |
| GetResult request | http://www.opengis.net/sos/2.0/GetResult | Y | Y | Y |
| GetResult response | http://www.opengis.net/sos/2.0/GetResultResponse | N | N | Y |
| a Although some values listed in the column appear to contain spaces, they shall not contain spaces.  NOTE The action URIs for the messages defined by the SWE Service Model and WS-Notification are not listed here – they can be found in Table 35 of [OGC 09-001] and the according paragraphs of WS-Notification. | | | | |

### SOAP Message Body

When implementing the SOAP Binding Extension, the XML encoded requests and responses shall be transferred in the body of a SOAP message.

|  |
| --- |
| **Requirement** |
| **http://www.opengis.net/spec/SOS/2.0/req/soap/message-body** |
| 1. The body of the SOAP messages that represent requests and responses of operations defined in this standard shall be encoded as defined in . |

### Example

1. The example shows a *GetCapabilities* request enclosed by a SOAP 1.2 envelope.

<soap12:Envelope

xmlns:soap12=http://www.w3.org/2003/05/soap-envelope

xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-envelope/soap-envelope.xsd http://www.opengis.net/sos/2.0 ../sos.xsd"

xmlns:sos="http://www.opengis.net/sos/2.0"

xmlns:wsa="http://www.w3.org/2005/08/addressing"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ows="http://www.opengis.net/ows/1.1">

<soap12:Header>

<wsa:To>http:/my.organization.org/services/sos</wsa:To>

<wsa:Action>http://www.opengis.net/sos/2.0/GetCapabilities</wsa:Action>

<wsa:ReplyTo>

<wsa:Address>http://my.client.com/client/myReceiver</wsa:Address>

</wsa:ReplyTo>

<wsa:MessageID>http://my.client.com/uid/msg-0010</wsa:MessageID>

</soap12:Header>

<soap12:Body>

<sos:GetCapabilities>

<ows:Sections>

<ows:Section>serviceIdentification</ows:Section>

<ows:Section>serviceProvider</ows:Section>

<ows:Section>contents</ows:Section>

</ows:Sections>

</sos:GetCapabilities>

</soap12:Body>

</soap12:Envelope>

# Annex A – Abstract test suite (normative)

## SOS Core Tests

This section defines conformance tests for the SOS Core. All tests are defined in one conformance class defined below

### Conformance Class: SOS Core

There are dependencies on conformance classes of OGC Web Services Common 1.1 [OGC 06-121r3], OGC SWE Common Service Model [OGC 09-001], and ISO Geographic Information — Observations and Measurements [ISO 19156].

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/core** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/core | |
| Dependency | http://www.opengis.net/doc/IS/OWS/1.1/clause/A4.2 | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.1.1 | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.1.2 | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.7.1 | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.7.2 | |
| Dependency | urn:iso:ts:iso:19156:clause:A1 | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/request-service** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/request-service |
| Test purpose | Verify that the server checks and accepts the service parameter value “SOS”. |
| Test method | Send several valid operation requests to the SOS server and verify that the SOS server answers with correct responses. Send invalid operation request with missing service attribute and service attribute with incorrect value to SOS server and verify that the server responds with appropriate exceptions. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/request-version** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/request-version |
| Test purpose | For all SOS request types defined in this specification, verify that the version parameter is checked by the server and the value “2.0.0” is accepted. |
| Test method | Send one valid request per request type to the service and verify that the operation result is as expected. Send invalid operation request with missing version attribute and version attribute with incorrect value to SOS server and verify that the server responds with appropriate exceptions. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/mandatory-operations** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/ds |
| Test purpose | Check whether the mandatory operations *GetCapabilities*, *GetObservations* and *DescribeSensor* are supported by the service. |
| Test method | Execute a *GetCapabilities*, *DescribeSensor*, and *GetObservation* request. Verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-sections** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-sections |
| Test purpose | Check whether the service accepts section parameters as defined in clause of this specification and in clause 7 of OWS Common [OGC 06-121r3]. |
| Test method | Submit GetCapabilities operation requests containing various values and combinations of values of the section parameter. Verify that the server provides the correct response to each request. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-version** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-version |
| Test purpose | Check whether the service checks the AcceptVersions parameter of *GetCapabilties* requests and accepts the value “2.0.0”. |
| Test method | Send valid *GetCapabilities* request containing the AcceptVersions parameter with value “2.0.0” to the service and verify that the Capabilities document is returned. Send a GetCapabilities request with AcceptVersions parameter set to a value other than “2.0.0” which is not supported by the service (e.g. “99.0.0”) to the service and verify that an appropriate exception is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-ows** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-ows |
| Test purpose | Check that the service accepts *GetCapabilties* requests as defined in clause 7 of OWS Common [OGC 06-121r3]. |
| Test method | Send valid *GetCapabilities* request to service and verify that a valid Capabilities document is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-request-structure |
| Test purpose | Check whether the service accepts *GetCapabilities* requests as defined in Subclause of this specification. |
| Test method | Send valid *GetCapabilities* request to service and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-response |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-structure |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-contents-structure |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-observationoffering-structure |
| Test purpose | Check whether the service returns an instance of the Capabilities type as defined in the conceptual model of clause of this specification. |
| Test method | Send valid *GetCapabilities* request to service and verify that response is conformant to Capabilities model. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-response-version** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-version |
| Test purpose | To verify that the default version of the Capabilities is “2.0.0”. |
| Test method | Send valid *GetCapabilities* request to service without AcceptVersions parameter and verify that default version of the Capabilities returned is “2.0.0”. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-operation-listing** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-operation-listing |
| Test purpose | Check whether the service supports the operations listed in the OperationsMetadata section of the Capabilities document. |
| Test method | Send valid operation request for each operation listed in OperationsMetadata section of Capabilities document and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-conf-class-listing** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-conf-class-listing |
| Test purpose | To verify that the Capabilities document advertises conformance classes which are supported by the server in addition to the SOS Core conformance class. |
| Test method | Retrieve Capabilities document, more specifically the Capabilities document including the ServiceIdentification section, and verify that the server passes all conformance tests of the conformance classes listed in the Profile list of the ServiceIdentification section. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-property-inheritance-mechanism** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-property-inheritance-mechanism |
| Test purpose | Check whether the properties of the Contents element in the Capabilities are inherited for all ObservationOfferings and check whether properties are contained as described in of this specification. |
| Test method | Send a valid *GetCapabilities* request to the service and verify that the properties of the Contents element in the Capabilities response are inherited for all ObservationOfferings as described in of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-spatialFilter-listing** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-spatialFilter-listing |
| Test purpose | Check whether the spatial operators and operands are listed in the FilterCapabilities section of the Capabilities. |
| Test method | Get a Capabilities document from the service and check that the spatial operators and operands are listed in the FilterCapabilities section of the Capabilities. Execute a *GetObservation* request for each combination of spatial operators and operands listed in the FilterCapabilities. Verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/spatial-filter-minimum** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/spatial-filter-minimum |
| Test purpose | Check whether the spatial operator BBOX is listed in the FilterCapabilities of the Capabilities document and whether it is supported for the *GetObservation* operation. |
| Test method | Query a Capabilities document from the service and check whether the BBOX operator is listed in the FilterCapabilities section. Send a *GetObservation* request containing the BBOX filter to the server and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-temporalFilter-listing** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-temporalFilter-listing |
| Test purpose | Check whether the temporal operators and operands are listed in the FilterCapabilities section of the Capabilities and are supported by the server. |
| Test method | Query a Capabilities document from the service and check whether the temporal operators and operands are listed in the FilterCapabilities section of the Capabilities. Execute a *GetObservation* request for each temporal operator and operand listed in the FilterCapabilities. Verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/temporal-filter-minimum** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-temporal-filter-minimum |
| Test purpose | Check whether the temporal operators TEquals and TDuring are listed as temporalFilters as well as TimeInstant and TimePeriod as temporal operands in the FilterCapabilities of the Capabilities document and verify that these are supported for the *GetObservation* operation. |
| Test method | Query a Capabilities document from the service and check that the operators and operands above are listed in the FilterCapabilities section. Send *GetObservation* requests for each combination of temporal operators and operands and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-offerings-observations** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-offerings-observations |
| Test purpose | Check whether the observations that an offering provides information about have not been created by another procedure than the one that is stated by the offering. |
| Test method | Send *GetCapabilities* request and cache the Capabilities response. Send a *GetObservation* request for each offering and verify that the observations returned have not been created by another procedure than the one that is stated by the offering. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-contents** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-contents |
| Test purpose | Check whether a Capabilities response contains a Contents section, if it has been requested. |
| Test method | Send *GetCapabilities* request containing a Sections parameter with value “Contents” and verify that the Capabilities response contains the Contents section. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-offering-identifier** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-offering-identifier |
| Test purpose | Check whether the server assigned a unique identifier to each ObservationOffering in the Capabilities. |
| Test method | Send *GetCapabilities* request containing a Sections parameter with value “Contents” to the server and verify that the server has assigned a unique identifier to each ObservationOffering in the Capabilities. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-response-format-om20** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-format-om20 |
| Test purpose | Check whether each ObservationOffering in the Capabilities contains at least one responseFormat parameter with value “http://www.opengis.net/om/2.0”. |
| Test method | Send *GetCapabilities* request containing a Sections parameter with value “Contents” to the server and verify that each ObservationOffering in the Capabilities contains at least one responseFormat parameter with value “http://www.opengis.net/om/2.0”. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/gc-exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/gc-exception |
| Test purpose | Check whether the server returns appropriate exception messages in case of an error while executing the operation. |
| Test method | Send invalid *GetCapabilities* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/ds** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/ds |
| Test purpose | Check that the service supports the *DescribeSensor* operations as specified in Section 11 of [OGC 09-001] and accepts such requests when their service property is set to “SOS” and their version property is set to “2.0.0” |
| Test method | Execute conformance tests as described in Subclause 20.1.7 of [OGC 09-001]. Verify that the service accepts valid DescribeSensor requests with service property set to “SOS” and version property set to “2.0.0” |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-request-structure |
| Test purpose | Check whether the service accepts *GetObservation* requests as defined in Subclause of this specification. |
| Test method | Send valid *GetObservation* request to service and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-parameters** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-parameters |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-omitting-parameters |
| Test purpose | Check whether the observations returned match the specified parameter values of the *GetObservation* request. |
| Test method | Send several *GetObservation* requests containing several valid parameters and check whether the observations returned match the request parameters. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-default-response-format** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-default-response-format |
| Test purpose | Check whether the format of observations returned in the response is O&M 2.0 [ISO 19156]. |
| Test method | Send a *GetObservation* request without responseFormat parameter and check whether the response is a GetObservationResponse containing observations encoded as O&M 2.0. To check the observations, use the conformance tests described in section A.1 of [ISO 19156]. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-response-format** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-response-format |
| Test purpose | Check that the format of the *GetObservation* response is valid according to the responseFormat parameter defined in the request. |
| Test method | Send valid *GetObservation* requests for each responseFormat listed in the Contents section of the Capabilities and verify that the SOS responds with the correct response format. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-response-type** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-response-type |
| Test purpose | To check that the type of the *GetObservation* response is valid according to the type of the GetObservationResponse as defined in subclause of this specification. |
| Test method | Send valid *GetObservation* request to server and verify that the SOS responds with the correct response format. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-observation-duplicates** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-observation-duplicates |
| Test purpose | Check that there are not duplicate observations in a *GetObservation* response. |
| Test method | Send several *GetObservation* requests to the service with offering parameter targeting all offerings with the same procedure but also targeting all observations and verify that there are no observation duplicates in the response. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/core/go-exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/core/go-exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *GetObservation* operation. |
| Test method | Send invalid *GetObservation* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |

## SOS Enhanced Operations Extension Tests

This section contains the conformance classes for the SOS Enhanced Operations Extension.

### Conformance Class: SOS Feature of Interest Retrieval

This conformance class defines conformance tests for the retrieval of features of interest.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/gfoi** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/gfoi | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gfoi/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/request-structure |
| Test purpose | Check that the service accepts *GetFeatureOfInterest* requests as defined in Subclause of this specification. |
| Test method | Send valid *GetFeatureOfInterest* request to service and verify that the server sends appropriate responses as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gfoi/parameters** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/parameters |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/omitting-parameters |
| Test purpose | Check that the features returned match the specified parameter values of the *GetFeatureOfInterest* request. |
| Test method | Send several *GetFeatureOfInterest* requests containing several valid parameters and verify that the features returned match the request parameters. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gfoi/spatial-filter** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/spatial-filter |
| Test purpose | Check that the spatialFilter of a *GetFeatureOfInterest* request is applied to the spatial property of the features of interest of observations and not of the observations themselves. |
| Test method | Send a *GetFeatureOfInterest* request containing a spatialFilter with a valueReference defining a spatial property of the features of interest. Verify that the geometries of the spatial property of the features which are returned by the server match the spatial filter. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gfoi/response-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/response-structure |
| Test purpose | Check that the structure of the response is conformant to the structure defined in Subclause of this specification. |
| Test method | Send a *GetFeatureOfInterest* request to the service and check that the response is returned as defined in Subclause 9.1.1.2 of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gfoi/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *GetFeatureOfInterest* operation. |
| Test method | Send invalid *GetFeatureOfInterest* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |

### Conformance Class: SOS Observation Retrieval By ID

This conformance class defines conformance tests for the retrieval of observations by ID.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/obsByIdRetrieval** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/obsByIdRetrieval | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gobi/request-structure |
| Test purpose | Check that the service accepts *GetObservationById* requests as defined in Subclause of this specification. |
| Test method | Send valid *GetObservationById* request to service and verify that the server sends an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/response-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gobi/response-structure |
| Test purpose | Check that the structure of the response is conformant to the structure defined in Subclause of this specification. |
| Test method | Send a *GetFeatureOfInterest* request to the service and check that the response is returned as defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/response-behavior** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gobi-response-behavior |
| Test purpose | Check that the server returns appropriate observations containing gml:identifier values as passed in the request. |
| Test method | Send valid *GetObservationById* request to the server and verify that the server returns observations containing gml:identifier values as passed in the request. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gobi/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *GetObservationById* operation. |
| Test method | Send invalid *GetObservationById* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/gobi/exception-no** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gobi/exception-no |
| Test purpose | Check that the server returns an exception message with ExceptionCode “InvalidParameterValue” and locator value “observation” if no observation was found matching one or more of the identifiers provided in the request. |
| Test method | Send *GetObservationById* with identifier value that is not associated with an observation hosted by the service and verify that an according exception message is returned as defined above. |
| Test type | Conformance |

## SOS Transactional Extension Tests

This section contains the conformance classes for the SOS Transactional Extension. It comprises four conformance classes which are defined below.

### Conformance Class: SOS Insertion Capabilities

This conformance class defines conformance tests for the InsertionCapabilities section in the Capabilities of the service.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/insertionCap** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/insertionCap/structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/insertionCap/structure |
| Test purpose | Check that the InsertionCapabilities is structured as defined in Subclause of this specification. |
| Test method | Query Capabilities of the service and check that InsertionCapabilities are valid according the model defined in Subclause 10.1.1 of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/insertionCap/capabilities-inclusion** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/insertionCap/capabilities-inclusion |
| Test purpose | Check that the InsertionCapabilities is contained in a Capabilities response, if it is requested. |
| Test method | Send *GetCapabilities* request with Sections parameter containing value “InsertionCapabilities” and verify that InsertionCapabilities are contained in response. |
| Test type | Conformance |

### Conformance class: SOS Sensor Insertion

This conformance class defines conformance tests for the insertion of sensors in the service.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/sensorInsertion** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/sensorInsertion | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.8 | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorInsertion/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/is/request-structure |
| Test purpose | Check that the service accepts an *InsertSensor* request as defined in Subclause of this specification. |
| Test method | Send a valid *InsertSensor* request to the service and verify that an appropriate response as defined in this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorInsertion/response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/is/supported-types |
| Test purpose | Check that the response is created as defined in Subclause |
| Test method | Execute the conformance test defined in Subclause 20.1.8 of [OGC 09-001] |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorInsertion/exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/is/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *InsertSensor* operation. |
| Test method | Send invalid *InsertSensor* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorInsertion/exception-unsupported-types** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/is/exception-unsupported-types |
| Test purpose | Check that an exception is returned with the ExceptionCode “InvalidParameterValue” and locator value “featureOfInterestType”, “observationType” or “resultType”, if one of the featureOfInterestType, observationType and resultType specified by the client in the SosInsertionMetadata element provided in the InsertSensor request are not supported by the SOS server (supported values are listed in the InsertionCapabilities section of the Capabilities document) |
| Test method | Send valid *InsertSensor* request containing an element of a type as described above with value that is not listed in the Capabilities to the server and verify that the server returns an exception with the ExceptionCode “InvalidParameterValue” and locator value “featureOfInterestType”, “observationType” or “resultType”. |
| Test type | Conformance |

### Conformance Class: SOS Sensor Deletion

This conformance class defines conformance tests for the deletion of sensors in the SOS.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/sensorDeletion** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/sensorDeletion | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | |
| Dependency | http://www.opengis.net/doc/IS/SWES/2.0/clause/A19.1.6 | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorDeletion/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ds/request-structure |
| Test purpose | Check that the service accepts *DeleteSensor* requests as defined in Subclause of this specification. |
| Test method | Execute conformance test defined in Subclause 20.1.6 of [OGC 09-001] |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorDeletion/obsoffering-deletion** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ds/obsoffering-deletion |
| Test purpose | Check that the ObservationOfferings of the sensor which has been deleted are no more listed in the Capabilities of the service. |
| Test method | Delete a sensor and query the Capabilities from the service. Check that the ObservationOfferings of the sensor are no longer listed in the Contents section of the Capabilities. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorDeletion/observation-deletion** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ds/observation-deletion |
| Test purpose | Check that the observations of the sensor which has been deleted are no more provided by the service. |
| Test method | Delete a sensor and send a *GetObservation* request for the sensor which has been deleted to the service. Verify that no observations but an Exception are returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/sensorDeletion/exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ds/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *DeleteSensor* operation. |
| Test method | Send invalid *DeleteSensor* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |

### Conformance Class: SOS Observation Insertion

This conformance class defines conformance tests for the insertion of observations in a SOS instance.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/obsInsertion | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | |
| Dependency | urn:iso:ts:iso:19156:clause:A1 | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/request-structure |
| Test purpose | Check that the service accepts *InsertObservation* requests as defined in Subclause of this specification. |
| Test method | Send a valid *InsertObservation* request to the service and verify that an appropriate response according to this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/supported-types** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/supported-types |
| Test purpose | Check that the observationTypes and resultTypes listed in the Capabilities are supported by the SOS and are also listed in the ObservationOffering to which the observations are added. |
| Test method | Query the Capabilities. Send valid InsertObservation requests for each constellation of observation- and resultTypes to the service and verify that an appropriate response according to this specification is returned. Retrieve a new Capabilities document and check that the observation- and resultTypes are also listed in the ObservationOfferings to which the observations have been added. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/property-constellation** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/request-structure |
| Test purpose | To verify that a particular constellation of procedure, observedProperty and ObservationOffering, observations is always encoded in the same observation- and resultType. Verify that the service checks whether the observation- and resultType is valid for a particular constellation. |
| Test method | Send an *InsertObservation* request with an incorrect observation- and resultType to the service. Verify that the observations are not inserted and an Exception is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/multiple-offerings** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/multiple-offerings |
| Test purpose | Check that the observations are inserted in all offerings which are associated with the sensor that has created the observations. |
| Test method | Send an *InsertObservation* request to the server and verify that the observations are inserted in all offerings which are associated with the sensor that has created the observations. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/response-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/response-structure |
| Test purpose | Check that the service returns an *InsertObservation* response as defined in Subclause of this specification. |
| Test method | Send a valid *InsertObservation* request to the service. Verify that service returns a response as defined in Subclause 10.4.1.2 of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/exception** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *InsertObservation* operation. |
| Test method | Send invalid *InsertObservation* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/exception-supported-types** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/exception-supported-types |
| Test purpose | Check that an exception is returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, if the observationType and resultType of the observation which should be inserted is not supported by the SOS or the types are not listed in the ObservationOfferings of the sensor in the Capabilities. |
| Test method | Send valid *InsertObservation* request containing an observation of invalid type to the server and verify that the server returns appropriate exception messages as described above. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/obsInsertion/exception-property-constellation** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/io/exception-property-constellation |
| Test purpose | Check that an exception is returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, if an observationType or resultType (or result structure / encoding) is inserted for the same constellation of procedure, observedProperty as well as ObservationOffering and that observationType/resultType is different than in previous insertions of observations with that property constellation. |
| Test method | Send valid *InsertObservation* request containing an observation of invalid type to the server and verify that the server returns appropriate exception messages as described above. |
| Test type | Conformance |

## SOS Result Handling Extension

This section describes conformance tests for the Result Handling Extension of the SOS.

### Conformance Class: Result Insertion

This conformance class defines conformance tests for result insertion in a SOS.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/resultInsertion | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/req/insertionCap | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/request-structure |
| Test purpose | Check whether the service accepts *InsertResultTemplate* requests as defined in Subclause of this specification. |
| Test method | Send a valid *InsertResultTemplate* request to the service and verify that an appropriate response according to this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/supported-types** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/supported-types |
| Test purpose | Check whether the observationTypes, resultTypes and resultEncodings listed in the Capabilities are supported for the SOS and are also listed in the ObservationOffering to which the observation results are added. |
| Test method | Query the Capabilities. Send valid *InsertResultTemplate* requests for each constellation of observation- and resultTypes to the service and verify that an appropriate response according to this specification is returned. Retrieve a new Capabilities document and check whether the resultTypes are also listed in the ObservationOfferings to which the results are added. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/property-constellation** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/property-constellation |
| Test purpose | For a particular constellation of procedure, observedProperty and ObservationOffering, results shall always be encoded in the same resultType and -Encoding. Verify that the service checks whether a resultType has already been inserted for a particular constellation. |
| Test method | Send two *InsertResultTemplate* requests with different resultTypes for the same constellation of procedure, observedProperty and ObservationOffering to the service. Verify that the service returns an Exception as response to the second *InsertResultTemplate* call. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/obs-template-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/obs-template-structure |
| Test purpose | Verify that the server checks whether the observation that is provided by the client in the ResultTemplate has an empty om:phenomenonTime, om:resultTime and om:result. For the first two properties, the nilReason has to be set to the value ‘template’. The procedure, featureOfInterest and observedProperty of the observation template shall not be empty. |
| Test method | Send a valid *InsertResultTemplate* request to the server and check whether it responds with an appropriate response. Afterwards, send an invalid *InsertResultTemplate* request to the server which does not fulfil the conditions defined above and verify that the server returns an exception message. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/result-structure-phenomenonTime** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/result-structure-phenomenonTime |
| Test purpose | The resultStructure in the ResultTemplate shall have at least a swe:Time or swe:TimeRange component with definition property set to the value “http://www.opengis.net/sos/2.0/observation/phenomenonTime”. Verify that the server checks the existence of this component and that the value of this component is used by the service to populate the om:phenomenonTime property of the observation template for each new result block the client is going to insert via the *InsertResult* operation. |
| Test method | Send a valid *InsertResultTemplate* request to the server and check whether it responds with an appropriate response. Then, send *InsertResult* requests for the template and verify that the value of this component is used by the service to populate the om:phenomenonTime property of the observation template for each new result block (e.g. by retrieving the full observations just inserted via a GetObservation request with appropriate temporalFilter). Afterwards, send an invalid *InsertResultTemplate* request to the server which does not fulfil the conditions defined above and verify that the server returns an exception message. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/result-structure-resultTime** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/result-structure-resultTime |
| Test purpose | If the resultStructure in the ResultTemplate has a swe:Time or swe:TimeRange component with definition property set to the value “http://www.opengis.net/sos/2.0/observation/resultTime”, verify that the value of this component is used by the service to populate the om:resultTime property of the observation template for each new result block the client is going to insert via the *InsertResult* operation. |
| Test method | Send a valid *InsertResultTemplate* request containing a component definition as defined above to the server and check whether it responds with an appropriate response. Then, send *InsertResult* requests for the template and verify that the value of this component is used by the service to populate the om:resultTime property of the observation template for each new result block (e.g. by retrieving the full observations just inserted via a GetObservation request with appropriate temporalFilter). |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/observation-time-provisioning** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/observation-time-provisioning |
| Test purpose | Verify that the service checks that a client encodes the om:phenomenonTime as a swe:Time or swe:TimeRange component with definition “http://www.opengis.net/sos/2.0/observation/phenomenonTime” in the resultStructure that it proposes to the service in the *InsertResultTemplate* operation request. |
| Test method | Send a valid *InsertResultTemplate* request containing a component definition as defined above to the server and check that it responds with an appropriate response. Then, send an invalid *InsertResultTemplate* request without the component defined above to the server and verify that an exception message is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/response-structure-irt** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/response-structure |
| Test purpose | Check that the service returns an InsertResultTemplateResponse as defined in Subclause of this specification. |
| Test method | Send a valid *InsertResultTemplate* request to the service. Verify that service returns a response as defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/exception-irt** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *InsertResultTemplate* operation. |
| Test method | Send invalid *InsertResultTemplate* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/exception-supported-types** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/exception-supported-types |
| Test purpose | Check that an exception is returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, if the observationType and resultType of the observation template which should be inserted is not supported by the SOS or the types are not listed in the ObservationOfferings of the sensor in the Capabilities. |
| Test method | Send invalid *InsertResultTemplate* request containing an observation of invalid type to the server and verify that the server returns appropriate exception messages as described above. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/exception-property-constellation** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/irt/exception-property-constellation |
| Test purpose | Check whether an exception is returned with the ExceptionCode “InvalidParameterValue” and locator value “observationType” or “resultType”, if an observationType or resultType (or result structure / encoding) is inserted for the same constellation of procedure, observedProperty as well as ObservationOffering and that observationType/resultType is different than in previous insertions of result templates with that property constellation. |
| Test method | Send invalid *InsertResultTemplate* request containing an observation template of invalid type to the server and verify that the server returns appropriate exception messages as described above. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/request-structure-ir** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ir/request-structure |
| Test purpose | Check that the service accepts *InsertResult* requests as defined in Subclause of this specification. |
| Test method | Send a valid *InsertResult* request to the service and verify that an appropriate response according to this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/resultValues-content** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ir/resultValues-content |
| Test purpose | The client shall encode the values of the observation result that is to be inserted via the *InsertResult* operation according to the resultStructure of the ResultTemplate it points to in the InsertResult request. The phenomenonTime and resultTime of the observation of which the result is going to be inserted shall be encoded in the result values (as swe:Time or swe:TimeRange component with definition value of “http://www.opengis.net/sos/2.0/observation/phenomenonTime” or “http://www.opengis.net/sos/2.0/observation/resultTime” respectively) so that the service can create a full observation from the information contained in the referenced template and the given result values. Verify that the server checks this requirement. |
| Test method | Send a valid *InsertResult* request to the service and verify that an appropriate response according to this specification is returned. Afterwards, send an *InsertResult* requests to the service where the resultValues are not encoded as defined for in the referenced result template (which defines both the structure and encoding of the values), and verify that the service responds with an appropriate exception message. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ir/response |
| Test purpose | Check that the service returns an InsertResultResponse as defined in Subclause of this specification. |
| Test method | Send a valid *InsertResultTemplate* request to the service. Verify that service returns a response as defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/exception-ir** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ir/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *InsertResult* operation. |
| Test method | Send invalid *InsertResult* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultInsertion/exception-unknown-result-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/ir/exception-unknown-result-structure |
| Test purpose | Check that an Exception is returned with exceptionCode value “InvalidParameterValue” and locator value “template”, if the structure and encoding of the inserted results does not adhere to the resultStructure and resultEncoding defined in the ResultTemplate inserted beforehand. |
| Test method | Send an invalid *InsertResult* request to the service containing results encoded in another structure and encoding as defined in the template inserted beforehand. Verify that that an Exception is returned with exceptionCode value “InvalidParameterValue” and locator value “template”. |
| Test type | Conformance |

### Conformance Class: Result Retrieval

This conformance class defines conformance tests for result insertion in a SOS.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/resultRetrieval | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/request-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/grt/request-structure |
| Test purpose | Check that the service accepts *GetResultTemplate* requests as defined in Subclause of this specification. |
| Test method | Send a valid *GetResultTemplate* request to the service and verify that an appropriate response according to this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/response-structure** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/grt/response-structure |
| Test purpose | Check that the service returns a GetResultResponse as defined in Subclause of this specification. |
| Test method | Send a valid *GetResultTemplate* request to the service. Verify that the service returns a response as defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/exception-grt** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/grt/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *GetResultTemplate* operation. |
| Test method | Send invalid *GetResultTemplate* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/request-structure-gr** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/request-structure |
| Test purpose | Check that the service accepts *GetResult* requests as defined in Subclause of this specification. |
| Test method | Send a valid *GetResult* request to the service and verify that an appropriate response according to this specification is returned. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/parameters** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/parameters |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/omitting-parameters |
| Test purpose | Check that the service returns all results that match the specified parameter values of the *GetResult* request. |
| Test method | Send several valid *GetResult* requests to the service and verify that the results returned match the specified parameter values of the *GetResult* request. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/response |
| Test purpose | Check whether the service response is returned as defined in Subclause of this specification. |
| Test method | Send a valid *GetResult* request to the service and verify that the response is valid according to the model defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/response-empty** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/response-empty |
| Test purpose | Check that the service response contains an empty resultValues property, if no observation matches the parameters of the *GetResult* request. |
| Test method | Send a valid *GetResult* request to the service containing parameters that do not match the observations provided by the SOS (e.g. by choosing a temporal filter that targets observations from the year 40000). Verify that the response contains an empty resultValues property. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/resultRetrieval/exception-gr** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/gr/exception |
| Test purpose | Check that the server returns appropriate exception messages in case of an error while executing the *GetResult* operation. |
| Test method | Send invalid *GetResult* requests to the server and verify that the server returns appropriate exception messages according to section of this specification. |
| Test type | Conformance |

## SOS Spatial Filtering Profile

This section describes conformance test for the spatial filtering profile.

### Conformance Class: Spatial Filtering Profile

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/spatialFilteringProfile** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/OMXML/2.0/conf/spatialObservation | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/spatialFilteringProfile/observationRestriction** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile/observationRestriction |
| Test purpose | Check that the response of a *GetObservation* request only contains observations which are conformant to the OM\_SpatialObservation defined in Section A.11 of [OGC 10-025]. |
| Test method | Send a valid *GetObservation* request to the service and verify that the observations returned are conformant to Spatial observation data defined in Section A.11 of [OGC 10-025]. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/spatialFilteringProfile/observedAreaRestriction** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/spatialFilteringProfile/observedAreaRestriction |
| Test purpose | Check that the observedArea of ObservationOfferings provided in the Capabilities document contain a geometry representing the boundary of the sampling geometries of observations that belong to that offering. |
| Test method | Query the Capabilities from the service. Afterwards, query the observations for each ObservationOffering from the service and check that the observedArea contains the boundary of these observations. |
| Test type | Conformance |

## SOS Binding Extension Tests

This section defines the conformance tests for the different bindings defined in the Binding Extension in section 14.

### Conformance Class: XML Encoding

This conformance class defines test for the XML Encoding requirements class.

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/xml-encoding** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/xml | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Dependency | http://www.opengis.net/spec/SWE/2.0/conf/xsd-simple-components | |
| Dependency | http://www.opengis.net/spec/SWE/2.0/conf/xsd-simple-encodings | |
| Dependency | http://www.opengis.net/spec/OMXML/2.0/conf/observation | |
| Dependency | http://www.opengis.net/spec/SWES/2.0/conf/XMLEncoding | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/xml/validation** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/xml/GeneralEncodingRules |
| Test purpose | Verify that XML implementations of the conceptual types defined in this specification are valid according to their XML Schema implementation. |
| Test method | For all XML instance documents received from the service or XML instance child elements that are in the namespace http://www.opengis.net/sos/2.0, verify that they are valid according to their XML Schema definition listed in .  Note: the sos.xsd can be used for validating any such XML instance against its schema definition. |
| Test type | Conformance |

### Conformance Class: KVP Binding Extension

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/kvp-core | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/url-encoded-values** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/url-encoded-values |
| Test purpose | Check that the server accepts URL encoded values. |
| Test method | Send a valid *GetCapabilities* KVP request to the service and verify that the server returns an appropriate response as specified in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/case-sensitivity** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/case-sensitivity |
| Test purpose | Verify that the server checks case-sensitivity of parameter values. |
| Test method | Send two *GetCapabilities* KVP requests to the service, one with a case-sensitive encoding of parameter values and one without (with then incorrect values). Check that the server responds with an Exception in the latter case. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/gc-request** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/gc-request |
| Test purpose | Check that the server accepts a *GetCapabilities* KVP request as defined in section . |
| Test method | Send a valid *GetCapabilities* KVP request to the service and verify that the server returns an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/gc-response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/gc-response |
| Test purpose | Check that the server returns a *GetCapabilities* response as defined in section . |
| Test method | Send a valid *GetCapabilities* KVP request to the service and verify that the server returns a valid Capabilities response. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/ds-request** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-request |
| Test purpose | Check that the server accepts a *DescribeSensor* KVP request as defined in section . |
| Test method | Send a valid *DescribeSensor* KVP request to the service and verify that the server returns an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/ds-time-encoding** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-time-encoding |
| Test purpose | Check that the server accepts validTime parameter values compliant to [ISO 8601]. |
| Test method | Send a valid *DescribeSensor* KVP request containing a validTime parameter value compliant to [ISO 8601] to the service and verify that the server returns an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/ds-response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/ds-response |
| Test purpose | Check that the server returns a *DescribeSensor* response as defined in section . |
| Test method | Send a valid *DescribeSensors* KVP request to the service and verify that the server returns a valid response. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/go-request** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-request |
| Test purpose | Check that the server accepts a *GetObservation* KVP request as defined in section . |
| Test method | Send a valid *GetObservation* KVP request to the service and verify that the server returns an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/go-BBOX-encoding** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-BBOX-encoding |
| Test purpose | Check that the BBOX parameter is encoded according to the definition in section . |
| Test method | Send a valid *GetObservation* KVP request containing a BBOX parameter to the service and verify that the server returns observations matching the BBOX of the request. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/go-temporalFilter-encoding** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-temporalFilter-encoding |
| Test purpose | Check that the temporalFilter parameter is used to filter on the phenomenonTime property of requested observations. The value of temporalFilter parameter shall be encoded compliant to [ISO 8601]. Instances and periods of time shall be supported. |
| Test method | Send a valid *GetObservation* KVP request containing a temporal filter as defined above to the service and verify that the server returns observations with phenomenonTime values that match the temporalFilter. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/go-response** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-core/go-response |
| Test purpose | Check that the server returns a *GetObservation* response as defined in section . |
| Test method | Send a valid *GetObservation* KVP request to the service and verify that the server returns a valid *GetObservation* response. |
| Test type | Conformance |

### Conformance Class: GetFeatureOfInterest KVP Binding

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/kvp-gfoi** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/core | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-core/gfoi-request** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi/gfoi-request |
| Test purpose | Check that the server accepts a *GetFeatureOfInterest* KVP request as defined in section . |
| Test method | Send a valid *GetFeatureOfInterest* KVP request to the service and verify that the server returns an appropriate response as defined in this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/kvp-gfoi/BBOX-encoding** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/kvp-gfoi/BBOX-encoding |
| Test purpose | Check whether the BBOX parameter is encoded according to the definition in section . |
| Test method | Send a valid *GetFeatureOfInterest* KVP request containing a BBOX parameter to the service and verify that the server returns features matching the BBOX of the request. |
| Test type | Conformance |

### Conformance Class: SOAP Binding

|  |  |  |
| --- | --- | --- |
| **http://www.opengis.net/spec/SOS/2.0/conf/soap** | | |
| Requirements | http://www.opengis.net/spec/SOS/2.0/req/soap | |
| Dependency | http://www.opengis.net/spec/SOS/2.0/conf/xml | |
| Dependency | http://www.opengis.net/spec/SWES/2.0/conf/SOAPBinding | |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/soap/exceptions** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/soap/exceptions |
| Test purpose | Check that the service returns exception codes defined by Subclause 8 in [OGC 06-121r3] and Subclause 15 in [OGC 09-001] and that the Exceptions are encoded as defined in Subclause 19.2 of [OGC 09-001]. |
| Test method | Send invalid SOAP requests to the service and verify that the server returns valid exception codes encoded as defined in Subclause 19.2 of [OGC 09-001]. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/soap/action-uris** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/soap/action-uris |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/soap/action-uris-exceptions |
| Test purpose | Check that the service accepts and returns valid action URIs as defined in Subclause of this specification. |
| Test method | Send valid SOAP requests to the service and verify that the server accepts and returns valid action URIs as defined in Subclause of this specification. |
| Test type | Conformance |
| Test | **http://www.opengis.net/spec/SOS/2.0/conf/soap/message-body** | |
| Requirement | http://www.opengis.net/spec/SOS/2.0/req/soap/message-body |
| Test purpose | Check that the service accepts and returns valid XML encoded requests and responses contained in the body of the SOAP messages. |
| Test method | Send valid SOAP requests for each supported operation to the service and verify that the service returns valid XML encoded responses. Validate the requests and responses with the schema defined in section . |
| Test type | Conformance |

# Annex B - Phenomena and Units of Measure (informative)

## B.1 Identifying and referencing Phenomena and Units of Measure

A critical issue for interoperability is defining a standard way to refer to the phenomena that are measured by sensors and the units of measure for those phenomena. This is important for both (i) discovery of SOS service instances in a catalog and (ii) to parameterize a request to a given service instance that offers a choice of observed properties. Because SOS is intended to be used in a wide variety of applications in a large number of application domains it is not feasible to construct a single comprehensive and authoritative dictionary for phenomena and units of measure. Observable phenomena include most properties of all feature types in all application domains (see O&M). The range of different phenomena and units of measure is large, unknown *a-priori*, and in fact both unknowable and incomputable. Phenomena and units of measure are often specific to a given domain and the mechanism used to reference them must support a decentralized approach.

One goal of SOS and SWE in general is to specify a standard mechanism for consistently identifying phenomena and units of measure that will scale (up or down) to handle any number of definitions in any application domain. The mechanism for identifying phenomena and units of measure must be flexible enough to handle this.

The solution for identifying phenomena and units of measure is to use external references. These may resolve to resources expressed in a variety of forms, utilizing various technologies including semantic web representations. GML dictionaries provide a relatively lightweight format which is compatible with the GML representation of data and web-based addressing patterns used by SOS. Services and clients use URIs to refer to specific entries in a particular GML dictionary.

The URI might be a URN in cases where the reference is to phenomena or units of measure that are defined by an OGC dictionary or a dictionary hosted by another well-known organization. URN values using OGC as the authority must follow the format specified in OGC document 06-023r1 - Definition identifier URNs in OGC namespace.

A URL may be used when the reference is to a new or non-standard definition, for example in the case that a service provides its own dictionary or uses a third-party dictionary that is not well-known. If the specific definition is a sub-element within a dictionary provided as a single resource, then its URL must include a fragment identifier or XPointer to locate the definition within the dictionary.

## B.2 Describing and defining Phenomena and Units of Measure

Entities like units of measure and phenomena are not physical objects in the real world. They are concepts and can only be defined by convention or by their relationship to other intangible concepts. Phenomena or units of measure that are defined in reference to other types can be considered to be derived or constrained entities and can be derived from more basic entities. Concepts like phenomena and units of measure occupy a different meta-level in the information modeling hierarchy, and their definitions are usually subject to more rigorous governance arrangements, compared with “instance” level data, such as observations and sensor instances. Hence, they will ideally be managed in a registry environment. The GML Dictionary representation may be thought of as a “static” view of such a collection of resources that would usually be provided by a service, such as a register or catalogue.

The SWE initiative relies on the existing GML support for identifying or defining units of measure. This is based on the usual hierarchy of base, derived and “other” (or “conventional”) units, such as defined by the Systeme International. The mechanism for deriving units is well-defined and can be done automatically using software as long as the base units are commonly understood. Ideally, though, UCUM symbols are used.

A GML conformant schema for describing phenomena derived by combination and/or constraining base phenomena was developed as part of the SWE initiative. This schema allows for the definition of base phenomena in much the same way that base units are defined in GML. Derived phenomena can be developed as a constraint on an existing phenomenon, an aggregation of existing phenomena, or as a composite of existing phenomena. SensorML 1.0.1 contains a schema for describing such phenomena.

# Annex C - Relationship to Other OGC Web Service Standards (informative)

This Annex describes the relationships of the SOS to other OGC service specifications which are also used to distribute geospatial data. Right now only the relationship to the Web Feature Service (WFS) is described.

## C.1 Relationship to Web Feature Service

The approach that has been taken in the development of SOS, and the SWE specifications on which it depends, is to carefully model sensors, sensor systems, and observations in such a way that the model covers all varieties of sensors and supports the requirements of all users of sensor data. SOS leverages the standard properties of these two data types (sensors, observations) to provide specialized operation signatures for observation data.

This may be contrasted with the approach taken in the Web Feature Service (WFS). WFS is based on a generic definition of a geographic feature that is flexible enough to encompass any real-world entity, and uses GML application schemas to define the feature type exposed by a specific service instance. Hence, the WFS “get data” request is highly parameterized since it must be fully generic. With this approach, interoperability requires organizations to agree on domain-specific GML application schemas. Clients that access a WFS for rich processing in a particular domain must have a-priori knowledge of the application schemas used in that domain.

The SOS defines a common model for all sensors, sensor systems and their observations. This model is “horizontal” since it applies to all domains that use sensors to collect data. The domain-specific details are encapsulated in the second layer (features-of-interest, observed properties, sensor descriptions) allowing the basic “observation” to be processed by a generic client. In that sense, the SOS also provides feature rich access to observation data and metadata.

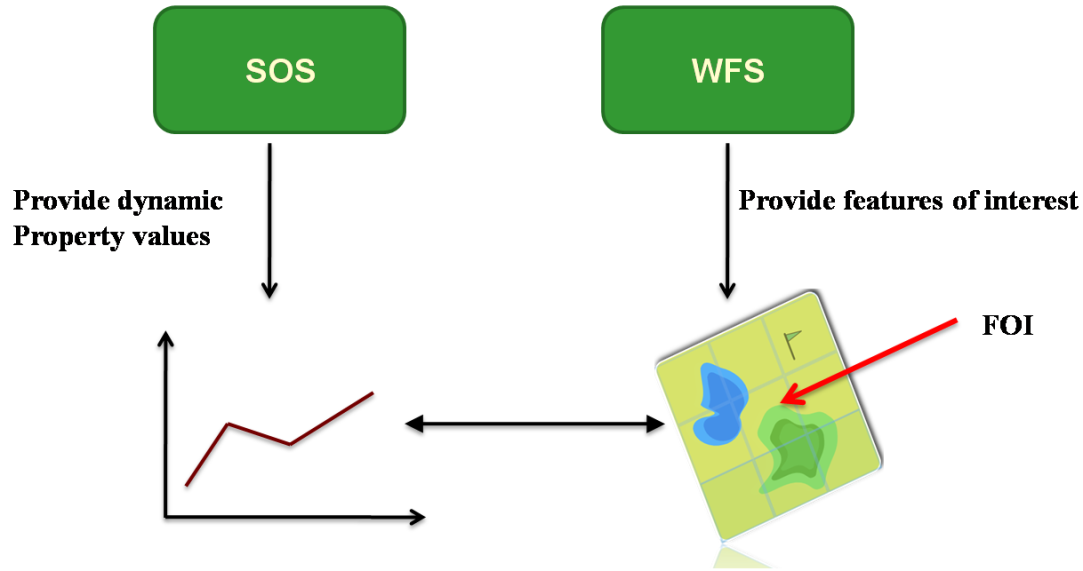


Figure 16‑1. Relationship between SOS and WFS where WFS is providing features of interest

Figure 16-1 and Figure 16-2 contain two simple examples how to couple SOS and WFS. In the first figure, the SOS is providing dynamic property values encoded in O&M observations for certain features of interest. These features are provided by an external WFS. For example, the SOS is providing surface temperature for a certain lake. Then the dynamic surface temperature values are provided by the SOS whereas the feature of interest, the lake, is provided by a WFS instance. As described above, the advantage for using an SOS is that it offers the surface temperature values and its metadata in the well-defined O&M format and that pre-defined filters (such as for time, location or producing procedure) can be used instead of the generic GetFeature operation of the WFS for retrieving the observations.

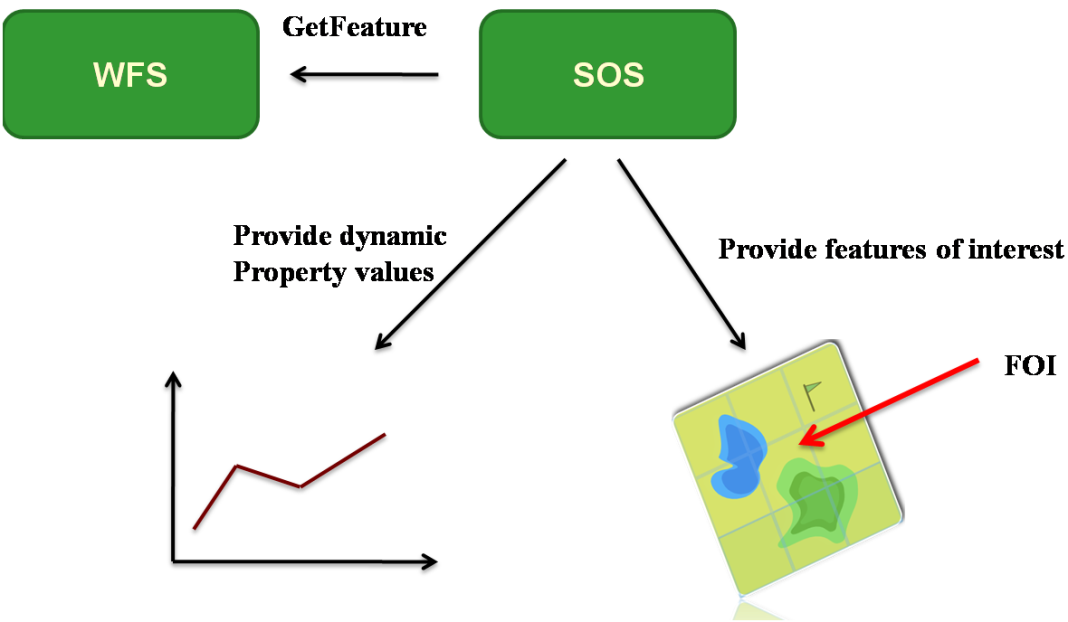


Figure 16‑2. Relationship between SOS and WFS where SOS is encapsulating WFS

Figure 2 shows another possibility for coupling SOS and WFS instances. In this case, the SOS is using a WFS at the backend for handling the features of interest. The SOS offers both the observations as well as the features of interest. In the example above, the client can retrieve the surface temperature observations as well as the feature of interest, the lake, from the same SOS instance. The operation for retrieving features of interest through the SOS interface is kept very simple, as there is only a limited set of query parameters.

The two examples above point out that it is not a question whether you want to use either SOS or WFS, but a question of how to combine or couple the two services. As stated above, the SOS describes a WFS profile with pre-defined observation feature types contained in the O&M specification and specialized operation signatures for retrieving these observations. Usually, it is used for providing dynamic property values for certain features of interest in the form of time series. These features of interest can be served either by the SOS instance itself (via GetFeatureOfInterest operation) or by an external WFS.

1. Krzysztof Janowicz, Arne Broering, Christoph Stasch & Thomas Everding (2010): Towards Meaningful URIs for Linked Sensor Data. In: Towards Digital Earth Workshop at Future Internet Symposium 2010 (CEUR Proceedings 640). 20th September 2010. Berlin, Germany [↑](#footnote-ref-1)
2. The SWE Service Model [OGC 09-001] is currently in the process of seeking approval as an official OGC standard. It is publicly available under this URL: http://portal.opengeospatial.org/files/?artifact\_id=40785 [↑](#footnote-ref-2)
3. A sensor system can be a simple thermometer, but can also consist of several sub-systems. So, a valid sensor system can be a system of sensors attached to a weather station, or it can be a network of spatially distributed sensors. [↑](#footnote-ref-3)
4. The design decision for specifying the multiplicity as “zero or many” instead of “zero or one” is that there may be many observable properties per ObservationOffering. Each one could use a different observation type or result type. However, a known issue is that the relation between observable property and observation / result type cannot be reflected in the Capabilities. [↑](#footnote-ref-4)
5. The design decisions for specifying those multiplicities are the following: The observableProperty, procedureDescriptionFormat, featureOfInterestType, and observationType are required as this information needs to be provided in the *InsertSensor* request and thus has to be provided even if no observations are assigned to the ObservationOffering, yet (and even if the *InsertSensor* operation is not implemented). The responseFormat is required as the requirement (**http://www.opengis.net/spec/SOS/2.0/req/core/gc-response-format-om20**)states that the default response format must be listed explicitly. The resultType depends on the observationType as stated in requirement (**http://www.opengis.net/spec/SOS/2.0/req/core/gc-observation-result-type**). [↑](#footnote-ref-5)
6. The design decision for specifying the multiplicity “zero or many” instead of “one or many” is that a client might want to request all observations for particular observedProperty (or featureOfInterest, procedure etc.). [↑](#footnote-ref-6)
7. A profile of this generic spatialFilter is given in Clause . This profile restricts the spatialFilter so that it is applied to the sampling location parameter of the observations. Those observations need to conform to the requirements class “Spatial Observation Data” defined in [OGC 10-025] Subclause 7.13. [↑](#footnote-ref-7)