Open Geospatial Consortium Inc.

Date: 2007-10-26

Reference number of this document: **OGC 06-009r6**

Version: 1.0

Category: OpenGIS® Implementation Standard

Editors: Arthur Na (IRIS Corp.), Mark Priest (3eTI)

Sensor Observation Service

Warning

This document is an OGC Standard. It is distributed for review and comment. It is subject to change without notice.

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.

Document type: OpenGIS® Publicly Available Standard
Document subtype: Implementation Standard
Document stage: Final
Document language: English

License Agreement

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

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

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

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

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

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable

Contents

Figures   Page

Tables   Page

i.   Preface

This document specifies the interface to a Sensor Observation Service, hereinafter "SOS". The SOS is one of a family of standards and specifications that make up the OGC Sensor Web Enablement activity, hereinafter "SWE". Currently, the other specifications that pertain to SWE are Sensor Model Language (SensorML), Observations and Measurements (O&M), Sensor Planning Service (SPS), Transducer Markup Language (TML), Sensor Alert Service (SAS), and Web Notification Service (WNS). Work on the predecessor to SOS, Sensor Collection Service (SCS), began during the OGC Web Services 1.1 (OWS 1.1) testbed initiative and focused mainly on the interoperable interface requirements for in-situ sensors and sensor networks. During the OWS 1.2 initiative, the SCS effort focused on the requirements for dynamic remote sensors (and sensor networks) and defined a more explicit interface employing SensorML. The SOS builds on previous SCS efforts and also incorporates an interface employing TML.

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

This specification deprecates previous versions including version 05-088.

ii.   Document terms and definitions

This document uses the specification 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 specification.

iii.   Submitting organizations

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

    a.   IRIS Corporation

b. 3eTI

c. University of Alabama in Huntsville

d. IFGI

e. York University

iv. Document contributor contact points

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

Contact	Company
Arthur Na (editor)	IRIS Corp.
Mark Priest (editor)	3eTI
Simon Cox	CSIRO
Mike Botts	UAH
Alexander Robin	UAH
Alexander C. Walkowski	IFGI
Ingo Simonis	IFGI
Johannes Echterhoff	IFGI
Steve Liang	York University
John Davidson	Image Matters
Harry Niedzwiadek	Image Matters

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
10/10/05	0.1.0	Arthur Na/ Mark Priest		Initial draft of SOS IPR, OWS 3, replacing Sensor Collection Service, OGC document 03-23r1
10/11/05	0.1.1	Harry Niedzwiadek	Section 6, Annexes A & C	Moved optional operators to Annex C. Updated Schema in Annex A. Fixed discrepancies in Section 6 or noted inconsistencies.
10/26/05	0.1.2	Arthur Na/ John Davidson	Section 5.3, Sections 8,9,10.	Added XML Spy notation. Changed parameter tables for all operations to four columns.
10/28/05	0.1.3	Mark Priest	Sections 6, 7, 9, Annex A & D	Extensive revisions were made. Added SOS Overview (Section 6), SOS Concept of Operations (Section 7), Transaction Profile

				(Section 9) and Future Considerations (Annex D). Also added normative schema for transaction operators to Annex A.
10/30/05	0.1.4	Harry Niedzwiadek	Section 10, Annex A	Review and edit of entire document. Moved enhanced operations from Annex C to Section 10, the main body of the document. Added normative schema for these operations in Annex A.
01/24/06	0.1.5	Alexander C. Walkowski	All	Some missing parameters added, as well as exceptions
05/20/06	0.1.6	Arthur Na	All	Corrected typographical, grammatical and other errors in text and examples.
01/12/07	1.0.0	Mark Priest	All	Address official comments and include lessons learned from the OWS-4 test bed.
01/31/07	0.1.2a	Simon Cox	All	Comments and corrections in review cycle.
02/01/07	0.1.2a	John Davidson	Appendix D	Editorial corrections; UML in Appendix D
02/26/07	0.1.2b	Alexander C. Walkowski	Section 5, 7, 8	Update of figures 7-4, 7-6, and 8-1 as well as some changes to improve the comprehensibility.
03/14/07	0.1.2b	Arthur Na	Section 7	Update of Figure 7-3 to correct errors.
10/19/07	1.0	Mike Botts, Johannes Echterhoff	throughout	Updated and corrected references to SWE specification versions and corrected schema and instances to reflect move from version 0.0 to 1.0
10/19/07	1.0	Mike Botts; Johannes Echterhoff	schema, Appendix A	corrected schema: version numbers, namespaces, schema locations, definition of temporal ops
10/26/07	1.0	Mike Botts; Johannes Echterhoff	throughout	Updated examples, updated references to OWSCommon specification
11/05/07	1.0	Mike Botts; Johannes Echterhoff	throughout	Fixed examples and schema using temporal filter elements beginning with 'T_' instead of 'TM_'
01/08/08	1.0	Johannes Echterhoff	Throughout	Updated examples and schema (namespaces + imports), updated figure D.1

The issues in this specification are captured in the following format:

> **Issue Name**: [Issue Name goes here. (Your Initials, Date)]
>
> **Issue Description:** [Issue Description.]
>
> **Resolution:** [Insert Resolution Details and History.] (Your Initials, Date)]

vi. Changes to the OpenGIS® Abstract Specification

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

vii. Future work

Further definitions and refinements of the optional enhanced and transactional operations are needed, along with examples of their use once they enter common usage. See Annex C for future work issues.

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

This version of the specification cancels and replaces all previous versions, including the Sensor Collection Service (SCS) Interoperability Program Report (IPR), Version 0.9.0, 03-023r1, and all previous versions of this document, in particular 05-088r2.

OGC 06-009r1 consists of the following parts, under the main body:

- Clause 1:   Scope
- Clause 2:   Conformance
- Clause 3:   Normative references
- Clause 4:   Terms and definitions
- Clause 5:   Conventions
- Clause 6:   Overview
- Clause 7:   Concept of Operations
- Clause 8:   Core Operations Profile
- Clause 9:   Transaction Operations Profile
- Clause 10:   Enhanced Operations Profile
- Annex A: XML Schemas (Normative)
- Annex B: SOS Instance Examples
- Annex C: Future Work: Issues to be Resolved
- Bibliography

Related standards and specifications

This specification was developed under the OWS3 initiative. The SOS Specification is part of the OGC Sensor Web Enablement activity and is related to the following SWE draft specifications.

- Observations and Measurements, OGC Document 07-022r1

- Sensor Alert Service, OGC Document 06-028r4

- Sensor Model Language (SensorML) , OGC Document 07-000

- Sensor Planning Service, OGC Document 07-014r3

- SWE Architecture, OGC 06-050r2

- Transducer Markup Language (TML), OGC Document 06-010r6

- Web Notification Service, OGC Document 06-095r1

Normative Annexes

Annex A is normative.

Informative Annexes

Annexes B and C are informative.

Introduction

The Open Geospatial Consortium's Sensor Web Enablement (SWE) activities, which have been executed principally through the OGC Web Services (OWS) initiatives under the Interoperability Program, is establishing the interfaces and protocols that will enable "Sensor Webs" through which applications and services will be able to access sensors of all types over networks such as the Internet and with the same standard technologies and protocols that enable the Web. These initiatives have defined, prototyped and tested several foundational components needed for a Sensor Web, namely:

1. **Observations & Measurements (O&M)** - The general models and XML encodings for sensor observations and measurements. O&M originated under OWS-1.1 and was significantly enhanced under the OWS-1.2, OWS-3, and OWS-4 testbed initiatives and is currently a Version 1.0 Implementation Specification.

2. **Sensor Alert Service (SAS)** – A service by which a client can register for and receive sensor alert messages. The service supports both pre-defined and custom alerts and covers the process of alert publication, subscription, and notification. This service is being defined as part of the Sensor Alert Service Interoperability Experiment begun in 2005.

3. **Sensor Model Language (SensorML)** – The general models and XML schema for describing sensors and processes associated with measurement. SensorML originated under NASA funding and was significantly enhanced under OWS-1.1, OWS-1.2, OWS-3, and OWS-4 testbed initiatives and is now available as a Version 1.0 Implementation Standard.

4. **Sensor Planning Service (SPS)** – A service by which a client can determine collection feasibility for a desired set of collection requests for one or more sensors/platforms, or a client may submit collection requests directly to these sensors/platforms. This service was defined in the OWS-1.2 testbed and enhanced in the OWS-3 and OWS-4 testbeds and is now available as a Version 1.0 Implementation Standard.

5. **Transducer Markup Language (TML) –** General characterizations of transducers (both receivers and transmitters), their data, how that data is generated, the phenomenon being measured by or produced by transducers, transporting the data, and any and all support data (metadata) necessary for later processing and understanding of the transducer data. This work was brought to OGC in the OWS-3 testbed and is now an OGC Version 1.0 Implementation Standard.

6. **Web Notification Service (WNS)** – A service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services. This service is useful when many collaborating services are required to satisfy a client request, and/or when significant delays are involved is satisfying the request. This service was defined under OWS 1.2 in support of SPS operations. WNS has broad applicability in many such multi-service applications.

This document specifies Sensor Observation Service. The other SWE components are specified under separate cover.

A Sensor Observation Service provides an API for managing deployed sensors and retrieving sensor data and specifically "observation" data. Whether from in-situ sensors (e.g., water monitoring) or dynamic sensors (e.g., satellite imaging), measurements made from sensor systems contribute most of the geospatial data by volume used in geospatial systems today. The general scenario for in-situ sensors is illustrated in Figure 1-1, where deployed sensors (Sn) of various types are grouped into several constellations (Cn) that are then accessed through some service (e.g., SOS).
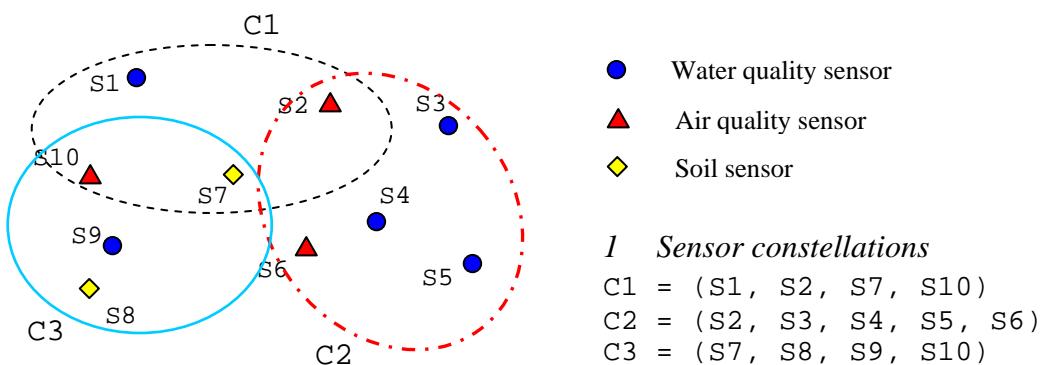
Figure 1-1 General Case for In-Situ Sensors

SOS has three mandatory "core" operations: **GetObservation, DescribeSensor**, and **GetCapabilities**. The **GetObservation** operation provides access to sensor observations and measurement data via a spatio-temporal query that can be filtered by phenomena. The **DescribeSensor** operation retrieves detailed information about the sensors and processes generating those measurements. The **GetCapabilities** operation provides the means to access SOS service metadata. Several optional, non-mandatory operations have also been defined. There are two operations to support transactions, RegisterSensor and InsertObservation, and six enhanced operations, including GetResult, GetFeatureOfInterest, GetFeatureOfInterestTime, DescribeFeatureOfInterest, DescribeObservationType, and DescribeResultModel.

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.

OpenGIS® Sensor Observation Service

1   Scope

This OpenGIS document describes the interface specification for the Sensor Observation Service.

2   Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3   Normative references

The following normative documents contain provisions which, 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.

OpenGIS® Abstract Specification, OGC document 99-100r1. Topic 0: Overview Topic 11 - Metadata (Same as ISO Metadata document 19115) Topic 12 - OGC Services Architecture
ISO 8601:1988(E), Data elements and interchange formats - Information interchange - Representation of dates and times.
Enhanced Filter Encoding, OGC document 05-093. Available to members online under Pending Documents.
Geography Markup Language (GML) Implementation Specification, Version 3.1.1, OGC document 03-105r1. Available to the public online under OpenGIS® Specifications.

Namespaces in XML. W3C Recommendation (14 January 1999). Available [Online]: <http://www.w3.org/TR/1999/REC-xml-names-19990114/>
Observations and Measurements, Version 1.0, OGC document 07-023r3. Available to the public online under OpenGIS® Specifications.
Sensor Model Language for In-Situ and Remote Sensors, Version 1.0, OGC document 07-000. Available to the public online under OpenGIS® Specifications.
Sensor Planning Service (SPS), Version 1.0, OGC document 07-014r3. Available to the public online under OpenGIS® Specifications.
Transducer Markup Language (TML), Version 1.0, OGC document 06-010r6. Available to the public online under OpenGIS® Specifications.
XML Schema Part 1: Structures. W3C Recommendation (2 May 2001). Available [Online]: <http://www.w3.org/TR/xmlschema-1/>
XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001). Available [Online]: <http://www.w3.org/TR/xmlschema-2/>
OpenGIS® Web Services Common Specification, OGC document 06-121r3. Available to the public online under OpenGIS® Specifications.

4   Terms and definitions

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

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

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

4.3   Observation
An act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property. A specialized event whose result is a data value.
[OGC 07-022r1]

4.4   Observation Offering

An observation offering is a logical grouping of observations offered by a service that are related in some way. The parameters that constrain the offering should be defined in

such as way that the offering is "dense" in the sense that requests for observations that are within the specified parameters should be unlikely to result in an empty set.

4.5 Phenomenon

Characteristic of one or more feature types, the value for which must be estimated by application of some procedure in an observation. [OGC 07-022r1]

4.6 Procedure

Method, algorithm or instrument. [OGC 07-022r1]

4.7 Result

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

4.8 Sensor

An entity capable of observing a phenomenon and returning an observed value. [OGC 07-000]

4.9 (Sensor) Platform

An entity to which can be attached sensors or other platforms. A platform has an associated local coordinate frame that can be referenced to an external coordinate reference frame and to which the frames of attached sensors and platforms can be referenced. [OGC 07-000]

5 Conventions

5.1 Symbols (and abbreviated terms)

API   Application Programming Interface

DCP   Distributed Computing Platform

CS-W   Catalog Service Web

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

SAS	Sensor Alert Service
SOS	Sensor Observation Service [formerly Sensor Collection Service, SCS]
SOS-T	Sensor Observation Service – transactional profile
SPS	Sensor Planning Service
SWE	Sensor Web Enablement
TML	Transducer Markup Language
UML	Unified Modeling Language
WNS	Web Notification Service
XML	eXtensible Markup Language

5.2   UML Notation

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

5.3   XML schema notation

Most diagrams that appear in this specification are presented using an XML schema notation defined by the XMLSpy[1] product and described in this subclause. XML schema diagrams are for informative use only, though they shall reflect the accompanied UML and schema perfectly.

The following sub-clauses provide an overview of the most commonly appearing elements of XML Spy diagrams. For a complete description, see

http://www.altova.com/manual2007/XMLSpy/SpyEnterprise/index.html?schemadesignview.htm.

5.3.1   Element

A named rectangle representing the most basic part of the XML Schema notation. Each represents an XML "Element" token. Each Element symbol can be elaborated with extra information as shown in the examples below.

---

[1] XML Spy: http://www.altova.com

Figure 5-1. Simple XML Spy element

This is a mandatory simple element. Note the upper left corner of the rectangle indicates that data is contained in this element.

5.3.2   Optional Element

Optional (non mandatory) elements are specified with dashed lines used to frame the rectangle.
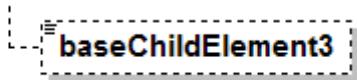
Figure 5-2. Optional XML Spy element

5.3.3   Recurring Element

This element (and its child elements if it has any) can occur multiple times.
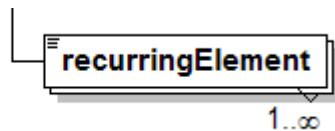
Figure 5-3. Recurring XML Spy element

This example shows a recurring element that must occur at least once but can occur an unlimited number of times. The upper bound here is shown with the infinity symbol.

5.3.4   Sequence Connector

The connection box, called a sequence indicator, indicates that the "SequenceElement" data is made up of three elements in a fixed order. In this example, the first two elements are mandatory and the third element is optional.
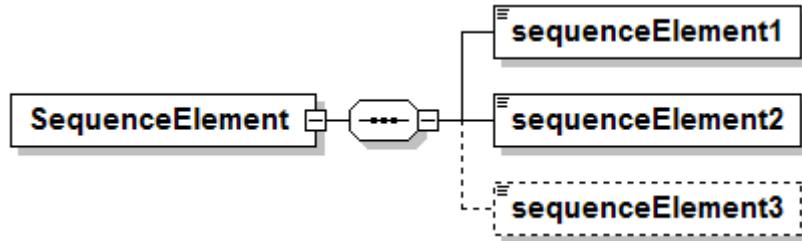
Figure 5-4. Sequence Connector in XML Spy

5.3.5   Choice Connector

The connection box here is a "choice" indicator, indicating that there is always going to be exactly one of the child elements listed on the right.
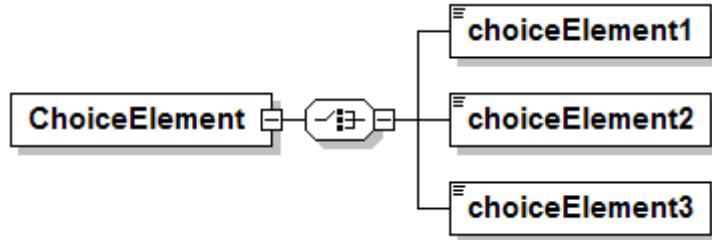
Figure 5-5. Choice Connector in XML Spy

5.3.6   Definition with Complex Type

This diagram illustrates the use of a complex type (i.e., "ex:AbstractElementType") in the declaration of an XML element (e.g., "AbstractElement").
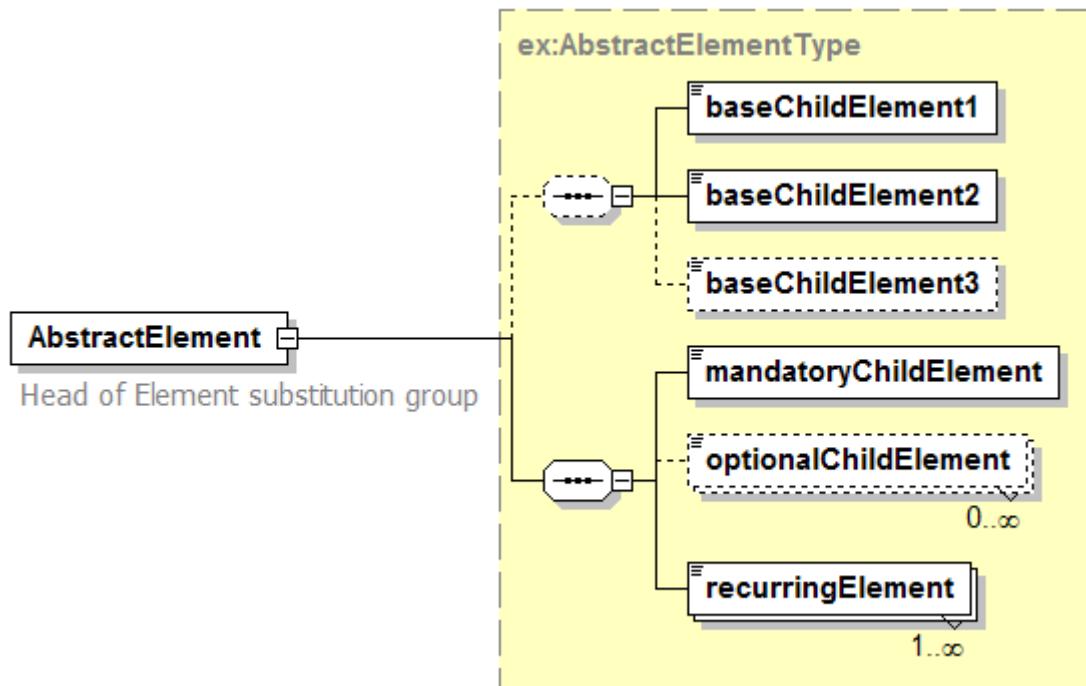
Figure 5-6. Definition with Complex Element in XML Spy

5.3.7 Complex Type

This diagram illustrates the definition of a complex type (i.e., "AbstractElementType"), extending another complex type (i.e.,"ex:BaseElementType") with three additional elements. Complex types describe a content model, and can be reused to specify that different elements have the same content model.
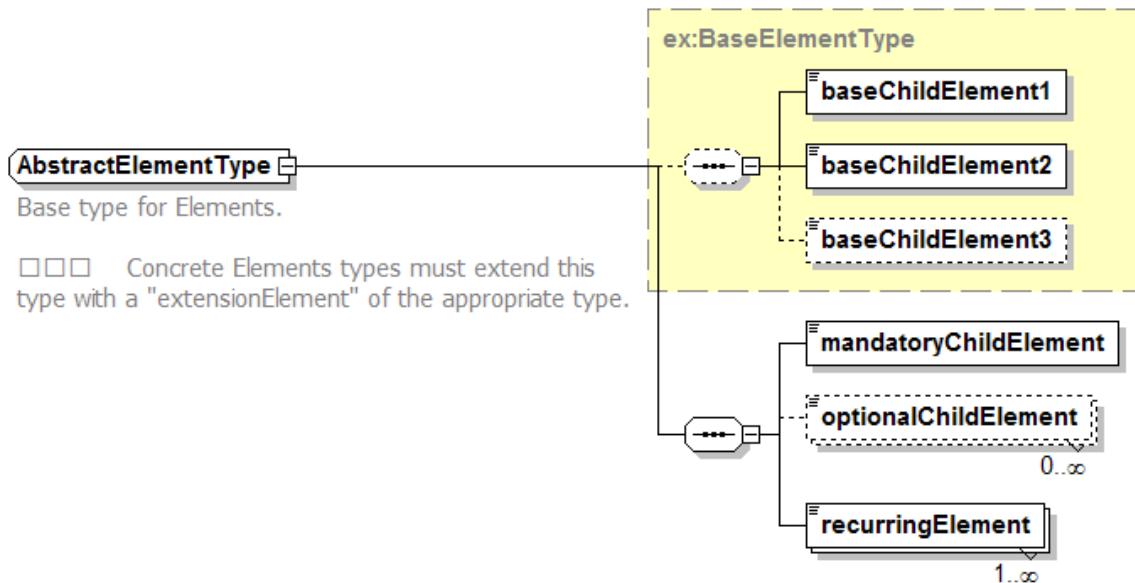
Figure 5-7. Complex Type in XML Spy

6   SOS Overview

6.1   General Approach

The goal of SOS is to provide access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems including remote, in-situ, fixed and mobile sensors. This is a challenging task because the users of sensor data have historically been divided into those who primarily deal with in-situ sensors and those who primarily deal with remote sensors. The terminology, perspective, and expectations of these two broad groups are different. SOS leverages the Observation and Measurements (O&M) specification for modelling sensor observations and the TransducerML and SensorML specifications for modelling sensors and sensor systems.

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 approach 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.

6.2   The Observation model

SOS is primarily designed to provide access to observations. A model for Observations and Measurements (O&M), based on patterns described by Fowler [1999] is formalized in an OGC Best Practice paper. The core model is summarized in Figure 6-1.
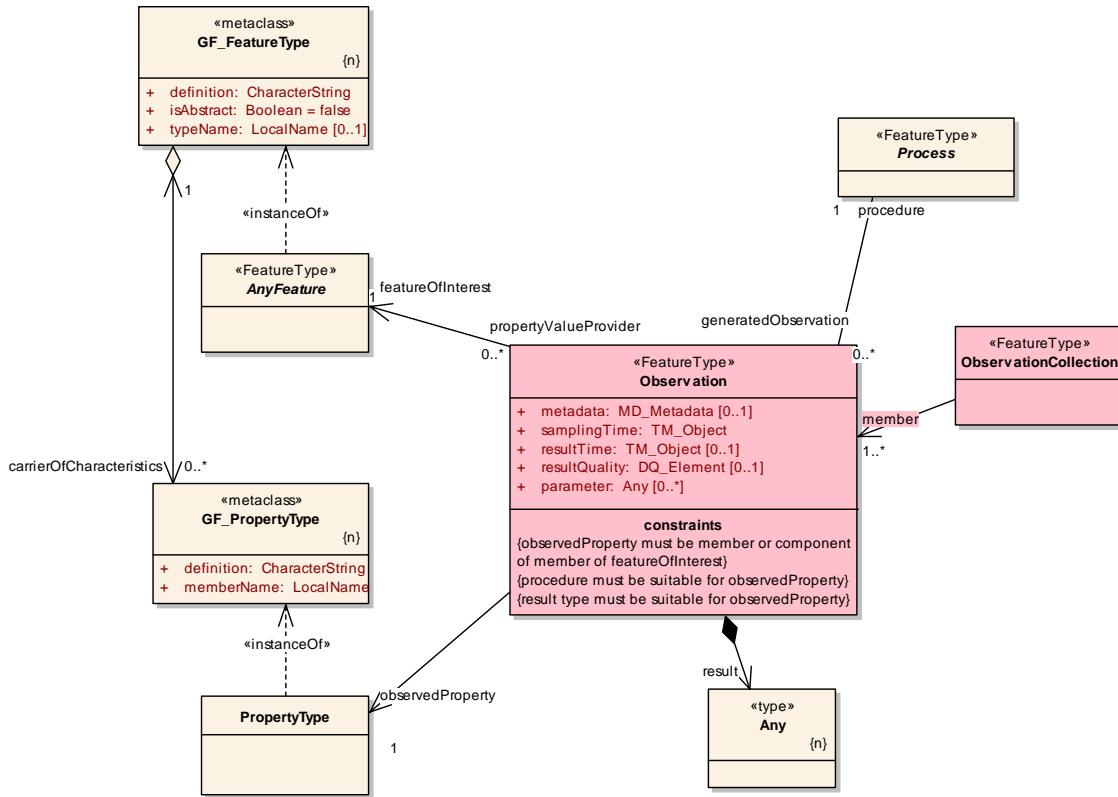
Figure 6-1. O&M Model Exract

The key to the model is that an Observation is modelled as an event which produces a result whose value is an estimate of a property of the observation target or feature of interest. An observation instance is classified by its *eventTime*, *featureOfInterest*, *observedProperty*, and the *procedure* used (often a Sensor).

6.3 Observation Offerings

An SOS organizes collections of related sensor system observations into Observation Offerings. The concept of an Observation Offering is often equivalent to that of a sensor constellation discussed in the introduction to this document. An Observation Offering is analogous to a "layer" in Web Map Service because each offering is typically a non-overlapping group of related observations. Each Observation Offering is constrained by a number of parameters including the following:

- Specific sensor systems that report the observations,

- Time period(s) for which observations may be requested (supports historical data),

- Phenomena that are being sensed,

- Geographical region that contains the sensors, and

- Geographical region that contains the features that are the subject of the sensor observations (may differ from the sensor region for remote sensors)

An SOS service instance should factor these classifiers into offerings such that in response to a GetObservation request the likelihood of getting an empty response for a valid query should be minimized.

For example, suppose an SOS instance advertises two sensors – one that reports wind speed and the other that reports air temperature. If these sensors are attached to the same weather station, they should probably be included in the same offering. That is because a GetObservation request for weather data for a given area that includes the weather station to which the sensors are attached and for time periods that the weather station is reporting will almost always have data for both sensors. If the client asks for wind speed only, air temperature only, or both, the time and location of the results should be the same.

On the other hand, if the two sensors are located on weather stations that are far apart in space or which report during non-overlapping time periods, then they should probably be factored into two distinct offerings. If they were put into the same offering, then the combinatorial space of that single offering would be relatively sparse. In other words, it would frequently be the case that a GetObservation request asking for temperature might return a result where one for wind speed might not. The client might have to make quite a few GetObservation requests on a "sparse" offering before finding data, which is clearly undesirable.

NOTE: ObservationOffering provides a mechanism for factoring groups of related observations within a single service instance. A functionally equivalent outcome could be obtained by factoring between different service instances.

6.4   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 very 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, utilising 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.

6.5   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 modelling 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. Alternatively, the UCUM symbols may be 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 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. The phenomena schema and encoding is described in O&M.

6.6 Filtering and Filter Expressions

The SOS GetObservation operation includes an ad-hoc query capability that allows a client to filter observations by time, space, sensor, and phenomena. This leverages the OGC Filter Encoding specification. However, the SOS augments the ad-hoc query capability with typed filter expressions bound to the standard observation properties. The GetObservation operation factors the spatial, temporal, and property-based query components into separate query elements. Instead of allowing a general filter expression, the filter is built up using a set of specific elements which have very clear semantics and restricted scope. This strategy is enabled by and is a corollary of the fact that SOS is designed to support a predefined information model (observations).

The capabilities document for SOS includes a FilterCapabilities section that is used to indicate the query parameters supported by the service. The filter capabilities element has been broken up into spatial capabilities, scalar capabilities, ID capabilities, and temporal capabilities. The factored design of the filter capabilities reflects the factored approach to query expressions in SOS.

This concept of defining strongly-typed filter expressions has been accepted as a change request to the Filter Encoding specification (CR 05-093 - Enhanced Filter Encoding) since it is expected to be useful in defining other service interfaces that are targeted at pre-defined information types.

7 SOS Concept of Operations

7.1 Overview

This Concept of Operations section is intended to outline a typical operational context in which an SOS is expected to be used. The intention is to provide a better understanding of the service and to help explain the operational context that was envisioned when the service was designed. This is only one view of the operational context and the SOS can be used in ways other than those described here. We expect that SOS will be used in ways not anticipated by the contributors to this specification which will help the specification to become more robust and useful than it is now. The Concept of Operations does not limit how an SOS can be used in an operational setting.

The Concept of Operations is divided into two sections. In the first section the perspective is that of a consumer of sensor observations. This is expected to be the most common perspective used with SOS. In the second section the perspective is that of a producer of sensor observations which is using the services of an SOS that supports the transactional SOS profile. The transactional profile allows sensor systems to be dynamically registered and allows sensor observations to be published to the service.

7.2   Concept of Operations for Sensor Data Consumer

7.2.1   Introduction

A sensor data consumer is interested in obtaining sensor observations from one or more sensors. At a high level the consumer might approach this problem from either a sensor-centric or an observation-centric point of view. A sensor-centric point of view would be used if the consumer was already aware of the existence of particular sensors and wanted to find observations for those sensors. An observation-centric point of view would be used if the consumer wanted to see sensor data from a particular geographic area and that capture particular phenomena but is not aware of any particular sensors a-priori.

In either case the consumer would perform service discovery using service capabilities information, usually via a catalog service, in order to find SOS service instances that can provide the desired sensor observations. After initial discovery the consumer could directly obtain observations from services or could perform additional discovery at the service level or get sensor metadata before obtaining sensor observations. Service-level discovery involves invoking the GetCapabilities operation to return information about the offerings that are available from each service. Detailed sensor metadata can be obtained after extracting the sensor system identifiers out of each observation offering by invoking the DescribeSensor operation.

Figure 9 shows the sensor data consumer in an operational context with OGC CS-W catalogs for services discovery and SOS service instances with observation offerings and observations. A service instance may talk directly to sensors or may be a proxy for other services as indicated in the diagram. Services can be organized into complex topologies using aggregation and other techniques but this is transparent to the data consumer. The consumer only needs to deal with registries and service interfaces.
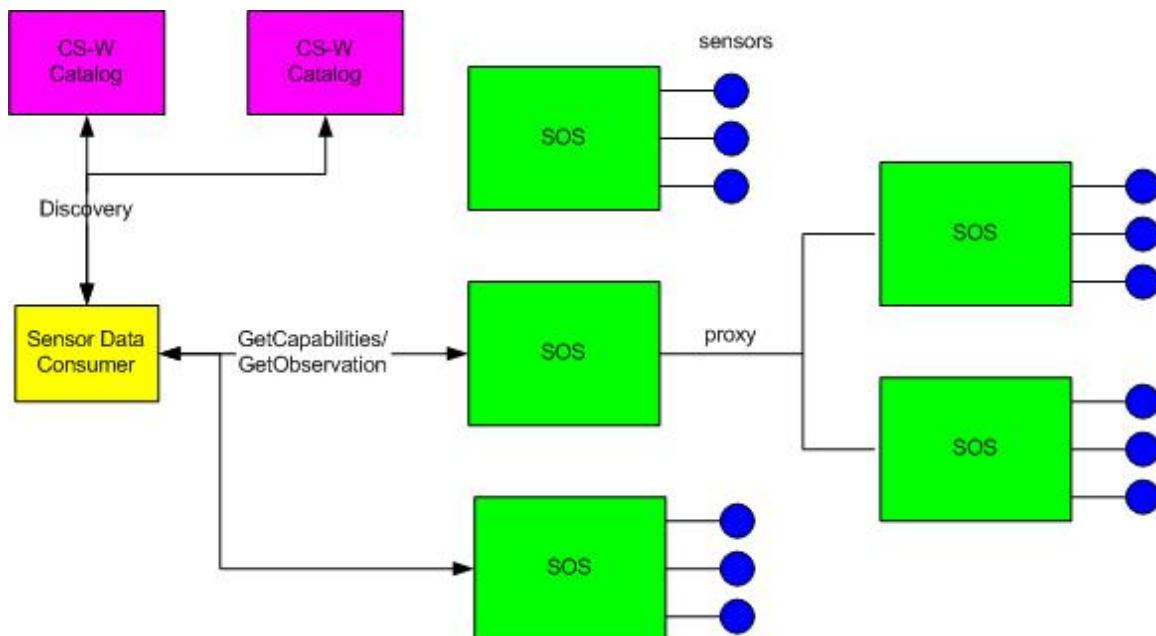
Figure 7-1 Sensor Data Consumer in Operational Context

Figure 10 is a flow chart that shows the mandatory and optional steps that can be followed by the sensor data consumer.
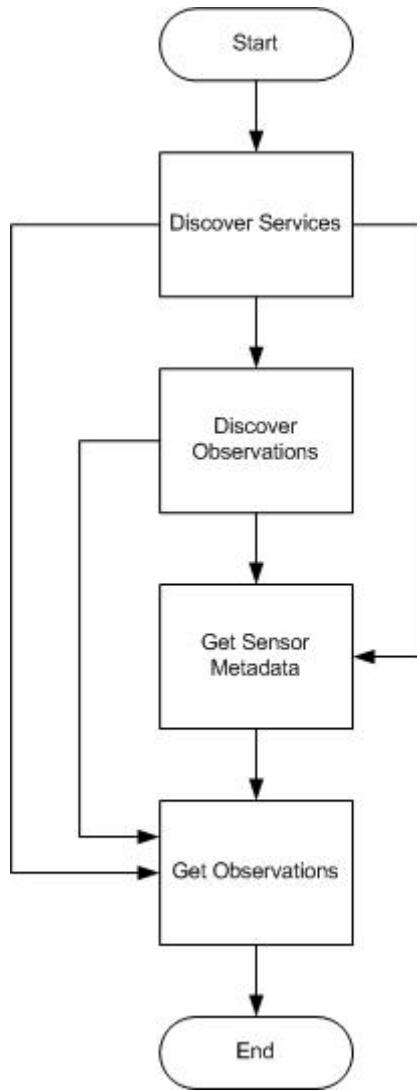
Figure 7-2 Sensor Data Consumer Flow Chart

Figure 11 is a sequence diagram that shows a sensor data consumer discovering two SOS instances from a CS-W catalog by using the GetRecords operation. The consumer then performs service-level discovery on each service instance by requesting the capabilities document and inspecting the observation offerings. The consumer invokes the DescribeSensor operation to retrieve detailed sensor metadata in SensorML or TML for sensors advertised in the observation offerings of the two services. Finally, the consumer calls the GetObservation operation to actually retrieve the observations from both service instances.
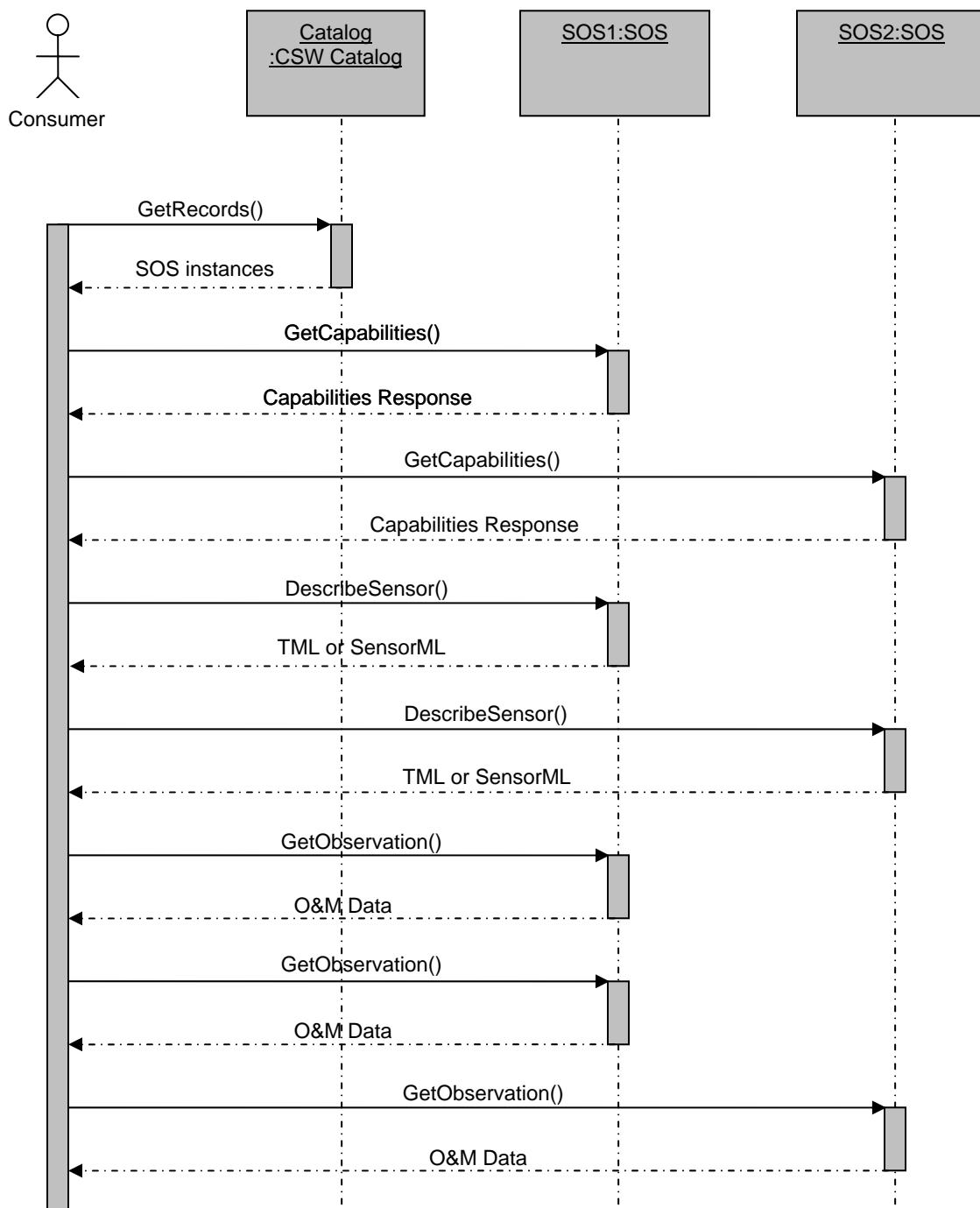
Figure 7-3 Sensor Data Consumer Sequence Diagram

NOTE: The GetCapabilities operation may be invoked directly by the end-user, as shown in this sequence. However, in a mature architecture the service capabilities are likely to be harvested into registers, where they are indexed and exposed through a discovery interface. The end-user then interacts with the register for discovery, and with the service itself only for access to observation data.

7.2.2 Service Discovery

Service discovery is accomplished by using one or more OGC Catalog Service (CS-W) instances. The CS-W specification is out of scope of this document but allows entities to be registered so that they can later be discovered by data consumers. It is also expected that catalogs may periodically harvest information from the capabilities documents of each service they know about in order to keep that service's attributes up to date. OGC Catalog Services allow the following information to be used for discovery:

- Time period of observations

- Phenomena captured by observations

- Spatial extent of observations

- GML names used in observation offerings

- GML descriptions used in observation offerings

OGC Catalogs may additionally enable discovery based on other parameters, such as the feature of interest, procedure/sensor, sampling rate.

7.2.3 Observation Discovery

Observation discovery happens at the service level. The service capabilities document includes detailed information about all of the offerings that are available from a service. This can be used to further target GetObservation requests. This is an optional step and can be skipped in many cases if the catalog response has sufficient information for the purpose of the consumer.

7.2.4 Get Sensor Metadata

Sensor metadata can be retrieved for any sensor that is advertised in an observation offering using the DescribeSensor operation. This will return a SensorML or TML document with detailed information about the sensor. This might be used to filter out observations produced by sensors that, for example, do not have robust enough error detection and correction or are not accurate enough for the purposes of the data consumer. This is an optional step that may be skipped in many cases.

NOTE: The available sensors are identified in the GetCapabilities response, so may also be harvested by a registry that is indexing the service.

7.2.5 Get Sensor Observations

Sensor observations are obtained using the GetObservation operation. This operation supports a selection mechanism that supports subsetting the observations that will be returned from a call to GetObservation. GetObservation allows the client to filter a large dataset to get only the specific observations that are of interest.

NOTE: The GetObservation operation is used for observation access and not generally for discovery, which is instead handled with ObservationOffer elements of the SOS Capabilities document and queries of online registry services.

7.3　Concept of Operations for Sensor Data Publisher

7.3.1　Introduction

The SOS includes a transactional profile with operations for inserting new sensors and sensor observations into an SOS. SOS instances that support the transactional profile are expected to be dependent on intermediary external services that talk with sensors rather than talking directly to sensors themselves. The intent is to support the development of commercial SOS products that can be dropped into any sensor network that includes sensor data producers that have a rudimentary SOS client capability.

The assumption is that sensor data producers are software entities running on resource constrained platforms that are attached to sensors – either physically or through a wireless connection of some kind. The producers have just enough processing power to generate simple XML insert documents that can be published to the transactional SOS. The transactional SOS is expected to be running on a capable server in a data center which can easily handle a large number of simultaneous connections from data producers.

Data consumers would request data from the transactional SOS using operations from the core profile and would not need to communicate with the ultimate data producers. Figure 12 shows sensor data producers in the operational context described above. The catalog still appears in the diagram but there is only one and it will often be co-located with the SOS service. The catalog would be used for discovery of the SOS instance if the data producers were capable enough to support a catalog client application. In some cases the sensor data publishers will need to be configured with the location of the SOS when they are deployed. The diagram is meant to represent a facility that is under the control of a single organization. Scalability would be achieved through aggregating a number of sensor data producers on the sensor side and by aggregating a number of SOS instances on the data center side for providing data to consumers.
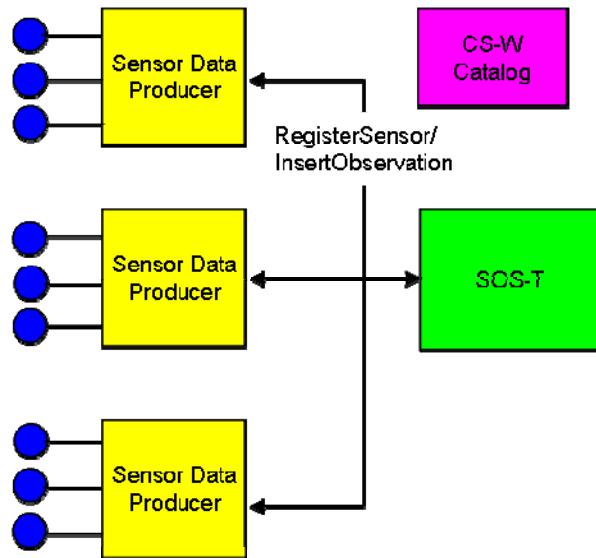
Figure 7-4 Operational Context for Sensor Data Producers

Figure 13 is a flow chart that shows the steps that can be followed by the sensor data producer. All of the steps in the flow are optional.
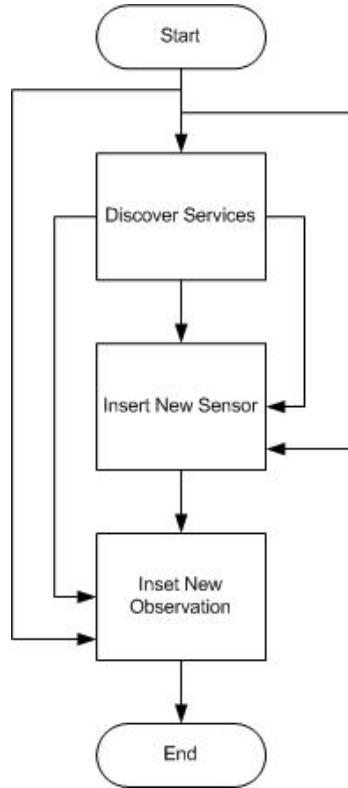
Figure 7-5 Sensor Data Producer Flow Chart

Figure 14 is a sequence diagram that shows how two sensor data producers interact with an SOS instance that supports the transactional profile.   The first producer discovers the SOS using a CS-W catalog. The second producer has been configured to know the location of the SOS. When the producers determine that they are communicating with new sensors they register those sensors with the SOS. The producers publish sensor observations from all sensors in their control to the SOS. The diagram also shows a sensor data consumer. Consumers can request observations from the SOS at any time using the GetObservation operation. The consumer does not need to be aware of the existence of the producers and vice versa.
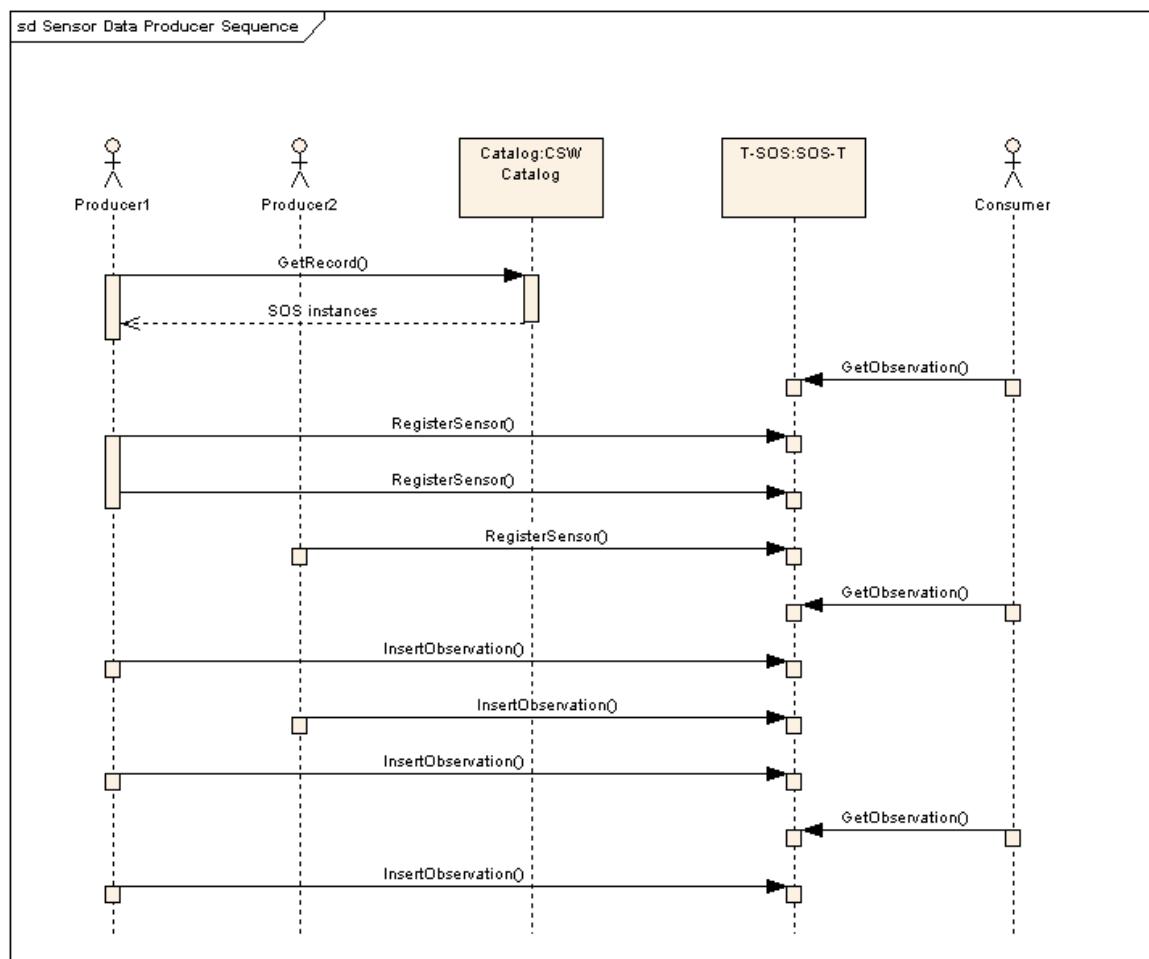
Figure 7-6 Sensor Data Producer Sequence Diagram

7.3.2 Service Discovery

Service discovery is done by using one or more OGC Catalog Service (CS-W) instances as described in section 7.2.1 above. We anticipate that sensor data publishers are likely to be more tightly bound to SOS instances than consumers and are less likely to incorporate a catalog client. Therefore, this step will probably be skipped for most producers and the producers will instead be manually configured with the location of the SOS instance when they are deployed.

7.3.3 Sensor Registration

A producer can only publish observations to an SOS if that SOS already knows about the sensor that generated the observations. The producer can look at the capabilities document of the SOS to determine whether the sensor is already known to the SOS. If not then the producer must register the sensor using the RegisterSensor operation.

7.3.4 Publishing of Observations

Once the producer has registered a sensor with an SOS instance it can begin publishing observations for that sensor. The SOS has the responsibility of packaging the observations into offerings and providing them to sensor data consumers.

8 Core Operations Profile

8.1 Overview

The SOS operations follow the general pattern of other OGC Web Services (OWS) and where appropriate inherit or re-use elements defined previously in the context of OWS. Key prior specifications include OWS Common [OGC 06-121r3] which provides the basic getCapabilities response, OGC Filter Encoding Specification (FES) [OGC 04-095] which provides a syntax for ad-hoc query formulation, and Web Feature Service (WFS) [OGC 04-094] which established the standard getCapabilities/describeX/getX interaction pattern for obtaining data and framing requests.

Operations defined for a Sensor Observation Service fall into four profiles: *core*, *enhanced*, *transactional* and *entire*. The mandatory *core* operations are **GetCapabilities**, **DescribeSensor** and **GetObservation** and are described in this clause. The *transactional* and *enhanced* operations are described in Clauses 9 and 10 respectively. The *entire* profile of SOS implements all of these operations (*core*, *enhanced* and *transactional*). The implementation and use of SOS operations is specified in accordance with the HyperText Transfer Protocol (HTTP) Distributed Computing Platform (DCP). Future versions may apply to other DCPs.

Clauses 8, 9, and 10 define the XML encodings for SOS operations. Clause 11 defines the KVP form of each of the operations.

8.2   GetCapabilities (mandatory)

8.2.1   Introduction

This operation allows clients to retrieve service metadata about a specific service instance. No "request" parameter is included, since the element name specifies the specific operation.

8.2.2   Request

The GetCapabilities operation request shall be as specified in Subclauses 7.2 and 7.3 of OWS Common [OGC 06-121r3]. The value of the "service" parameter shall be "SOS". The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Tables 6 and 10 of [OGC 06-121r3], with the additions listed in Table 1 below.

Table 1 — Additional Section name values and meanings

Section name	Meaning
Contents	Return the contents section in service metadata document that contains information about the observation offerings.
Filter_Capabilities	Return the filter capabilties section in service metadata document that contains information about the supported filters.

The "Multiplicity and use" column in Table 3 of [OGC 06-121r3] specifies the optionality of each listed parameter in the GetCapabilities operation request.   Annex B provides an example of a SOS GetCapabilities document.

8.2.3   Response

The response to a GetCapabilities operation is an XML encoded document. This document provides clients with service metadata about a specific service instance, including metadata about the tightly-coupled data served. If the server does not implement the updateSequence parameter, the server shall always return the complete Capabilities document, without the updateSequence parameter. When the server implements the updateSequence parameter and the GetCapabilities operation request included the updateSequence parameter with the current value, the server shall return this element with only the "version" and "updateSequence" attributes. Otherwise, all optional elements shall be included or not depending on the actual value of the Sections parameter in the GetCapabilities operation request.

Figure 8-1. GetCapabilities Response

The portions of the GetCapabilities response document that are defined in the OWS Common specification are not modified for SOS. The subsections that follow contain only the sections of the response that are specific for the SOS; namely these are the Filter_Capabilities and the Contents section.

8.2.3.1   FilterCapabilities Section

The FilterCapabilities section is used to indicate what types of query parameters are supported by the service. These capabilities refer to the parameters of the GetObservation operation which is the only operation that includes OGC filter-like expressions.

8.2.3.2   Contents Section

The concept of an Observation Offering is discussed in section 6.3 above. It is a way to organize sets of sensor observation groupings that can be queried using the

GetObservation operation. Each Observation Offering is constrained by a number of parameters described in the table below.

The Observation Offering element derives from the GML abstract feature type so it includes a GML Envelope that describes the geographic footprint of the offering. It also includes a gml:id and standard GML object metadata elements.
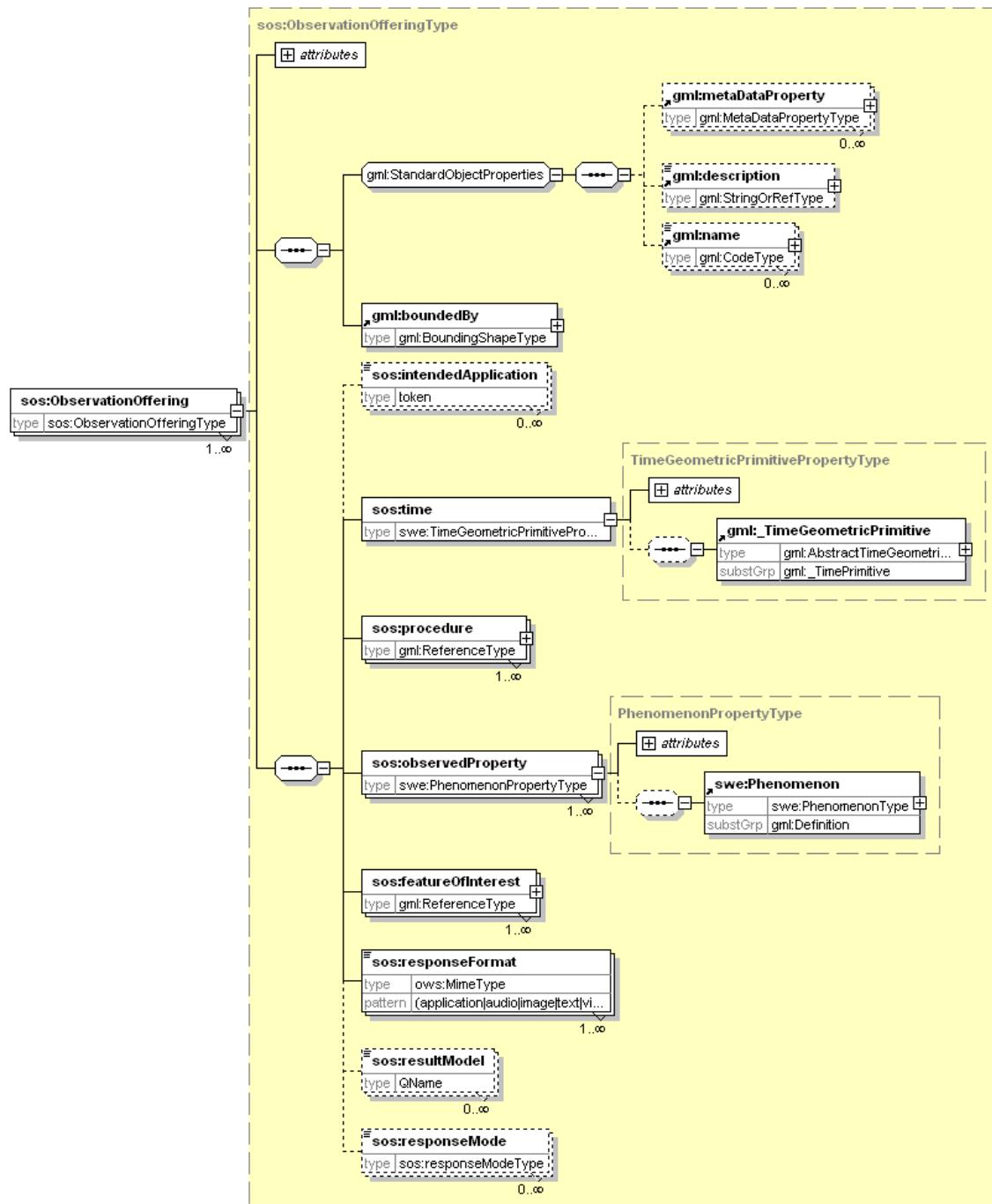
Figure 8-2 Observation Offering

Table 2 Observation Offering Constraints

Name [a]	Definition	Data type and values	Multiplicity and use
intendedApplication	The intended category of use for this offering such as homeland security or natural resource planning	Token	Optional
time	Time period for which observations can be obtained. This supports the advertisement of historical as well as real-time observations.	gml:TimeGeometricPrimitivePropertyType	One (mandatory)
procedure	A reference to one or more procedures, including sensor systems, instruments, simulators, etc, that supply observations in this offering. The DescribeSensor operation can be called to provide a SensorML or TML description for each system.	gml:ReferenceType	One or more (mandatory)
observedProperty	The observable/phenomenon that can be requested in this offering.	swe:PhenomenonPropertyType	One or more (mandatory)
featureOfInterest	Features or feature collections that represent the identifiable object(s) on which the sensor systems are making observations. In the case of an in-situ sensor this may be a station to which the sensor is attached representing the environment directly surrounding the sensor. For remote sensors this may be the area or volume that is being sensed, which is not co-located with the sensor. The feature types may be generic Sampling Features (see O&M) or may be specific to the application domain of interest to the SOS. However, features should include spatial information (such as the GML boundedBy) to allow the location to be harvested by OGC service registries.	gml:ReferenceType	One or more (mandatory)
responseFormat	MIME type of the data that will be returned as the result of a GetObservation request. This is usually text/xml; subtype="om/0.0.0". In the case that data is delivered out of band it might be text/xml;subtype="tml/2.0" for TML or some other MIME type.	ows:MimeType	One or more (mandatory)
resultModel	Indicates the namespace-qualified name of the result element that will be included in the document returned from a call to GetObservation for this offering. This element must be in the om:Observation substitution group and is typically om:Observation or a specialized extension.	QName	Optional
responseMode	Indicates what modes of response are supported for this offering. The value of resultTemplate is used to retrieve an observation template that will later be used in calls to GetResult. The other options allow results to appear inline in a resultTag (inline), external to the observation element (out-of-band) or as a MIME attachment (attached).	inline out-of-band attached resultTemplate	Optional
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

8.2.4   Exceptions

When a SOS server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 06-121r3]. The

allowed exception codes shall include those listed in Table 8 of Subclause 7.4.1 of [OGC 06-121r3], if the updateSequence parameter is implemented by the server.

8.2.5   Examples

See Annex B for an example of a GetCapabilities response document.

8.3   DescribeSensor (mandatory)

8.3.1   Introduction

One method of obtaining metadata that describes the characteristics of an observation procedure (sensor or sensor constellation) is to retrieve it from a catalog. However, a catalog may only contain high-level information about the observable properties, locations, contact information, etc. Due to the possibly voluminous amounts of sensor related metadata, the "describe" operations support retrieval of the highest level of detail about the platforms and sensors associated with an SOS.

DescribeSensor is designed to request detailed sensor metadata. Likewise, the response to a GetCapabilities request could provide a list of sensors associated with an SOS. Sensors are devices for the measurement of physical quantities. Both Sensor Model Language (SensorML) and Transducer Markup Language (TML) provide a detailed definition of a sensor for In-situ and Remote Sensors specification.

For the purpose of this discussion, there are two main kinds of sensor, in-situ and remote-dynamic. The DescribeSensor operation is useful for obtaining detailed information of sensor characteristics encoded in either SensorML or TML. The sensor characteristics can include lists and definitions of observables supported by the sensor. The SensorML and TML specifications detail the response model for the DescribeSensor operation.

8.3.2   Request

Figure 8-3 describes the syntax of the SOS DescribeSensor operation.
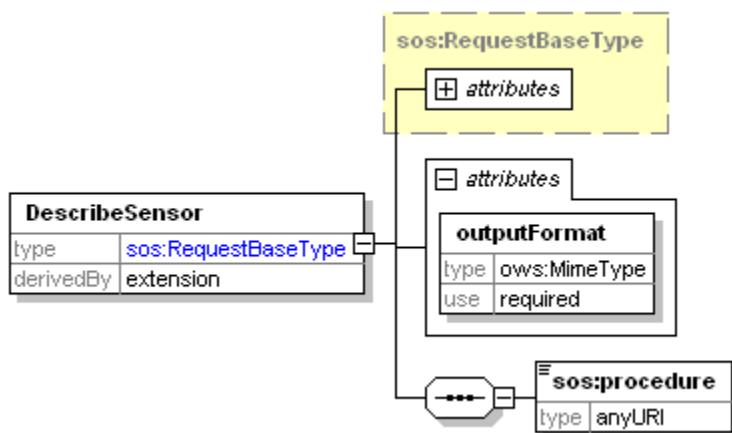
Figure 8-3 DescribeSensor Request

Table 3 Attributes of DescribeSensor Request

Name [a]	Definition	Data type and values	Multiplicity and use
outputFormat	The outputFormat attribute specifies the desired output format of the DescribeSensor operation. This can be a MimeType or QName for example. This is text/xml;subtype="sensorML/1.0.0" for SensorML or text/xml;subtype="TML/1.0" for TML.	ows:MimeType	Required
SensorId	The sensorId parameter specifies the sensor for which the description is to be returned. This value must match the value advertised in the xlink:href attribute of a procedure element advertised in the SOS GetCapabilities response.	anyURI	One (mandatory)
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS.	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

8.3.3   Response

A SensorML or TML document describing the sensor system.

8.3.4   Exceptions

When a SOS server encounters an error while performing a DescribeSensor operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

8.3.5   Examples

8.3.5.1   Example1

This example illustrates requesting information about a specific sensor instance. The request:

```
<DescribeSensor xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0  http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" outputFormat="text/xml;subtype=&quot;TML/1.0&quot;" version="1.0.0">
 <procedure>urn:ogc:object:Sensor:MyOrg:12349</procedure>
</DescribeSensor>
```

The response (abbreviated):

```
<tml:system xmlns:tml="http://www.opengis.net/tml">
    <tml:identification>
        <tml:uid>urn:ogc:object:Sensor:MyOrg:12349</tml:uid>
    </tml:identification>
    <tml:transducers>
        <tml:transducer>
            <tml:identification>
                <tml:uid>urn:ogc:object:Sensor:MyOrg:12349</tml:uid>
                <tml:description>GID3 Chemical Sensor</tml:description>
                <tml:manufacture>Smiths Detection</tml:manufacture>
                <tml:modelNumber>GID3</tml:modelNumber>
                <tml:serialNumber>333334444</tml:serialNumber>
            </tml:identification>
```

```
        <tml:transducerClass>
            <tml:transmitterReceiver>receiver</tml:transmitterReceiver>
            <tml:insituRemote>insitu</tml:insituRemote>

. . .

</tml:system>
```

8.4   GetObservation (mandatory)

8.4.1   Introduction

The GetObservation operation is designed to query a service to retrieve observation data structured according to the Observation and Measurement specification. Upon receiving a GetObservation request, a SOS shall either satisfy the request or return an exception report.

8.4.2   Request

A <GetObservation> message contains one or more elements that constrain the observations to be retrieved from a Sensor Observation Service. Each GetObservation query element has mandatory attributes of service and version. The mandatory version element attribute must correspond to the specific service interface version negotiated between the service and client during the service binding process. The required service attribute explicitly must be "SOS".
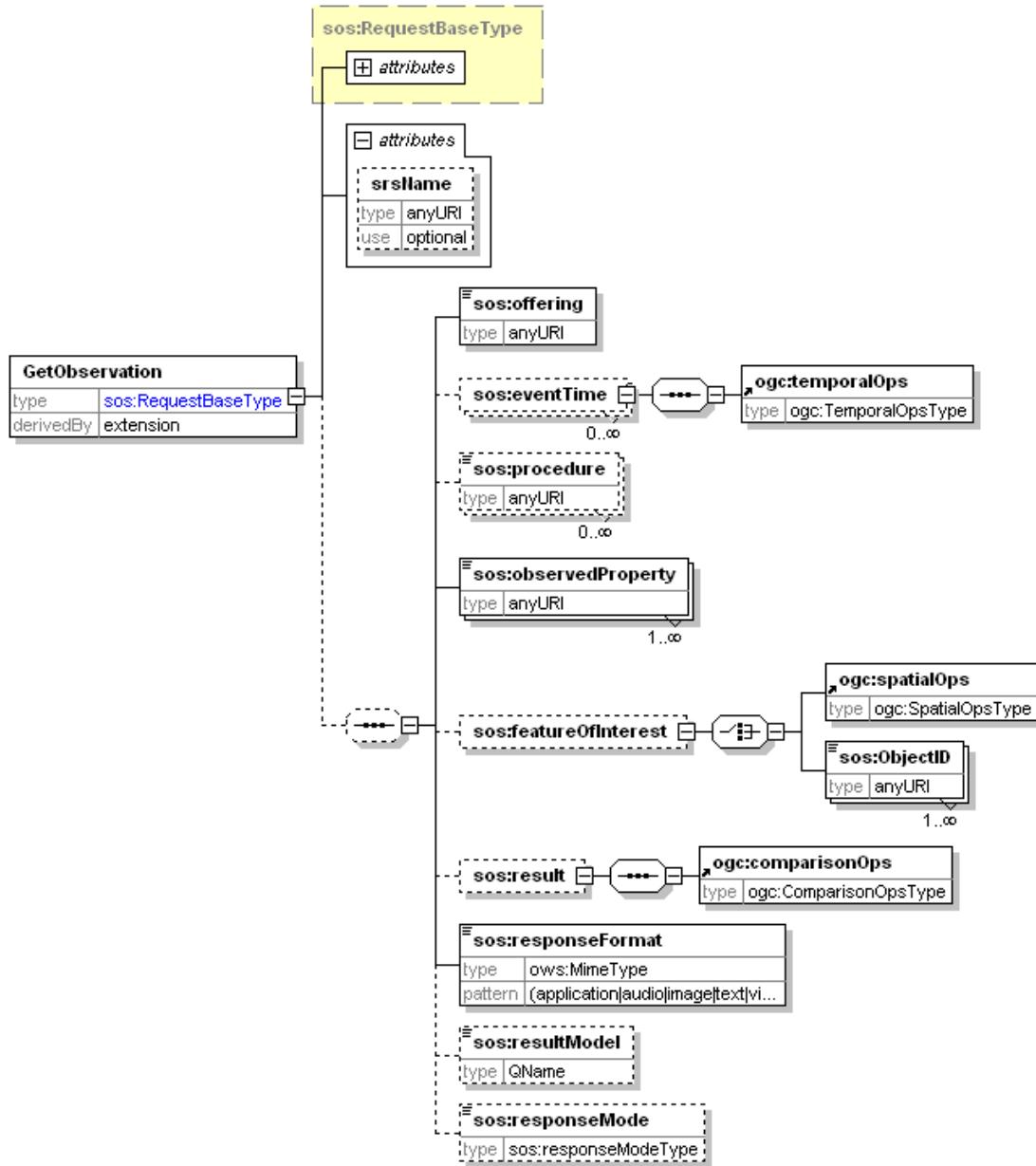
Figure 8-4 GetObservation Request

Table 4 Parameters of GetObservation Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Version of SOS supported.	One (mandatory)
srsName	Defines the spatial reference system that should be used for any geometries that are returned in the response. This must be one of the advertised values in the offering specified in gml:srsName elements.	anyURI	One (optional)
offering	Specifies the offering URI advertised in the GetCapabilities document. This must match the gml:name of the offering or be constructed as a URL with a fragment identifier resolving to the offering gml:id. All of the following parameters are dependent on the selected offering.	anyURI	One (mandatory)
eventTime	Specifies the time period(s) for which observations are requested. This allows a client to request observations from a specific instant, multiple instances or periods of time in the past, present and future. The supported range is listed in the selected offering capabilities.	ogc:temporalOps	Zero or many (Optional)
procedure	The procedure parameter specifies the sensor system(s) for which observations are requested.	anyURI	Zero or many (Optional)
observedProperty	Specifies the phenomenon for which observations are requested.	anyURI	One or many (mandatory)
featureOfInterest	Specifies the feature for which observations are requested. This can either be represented by a reference to a feature ID advertised in the capabilities document or can be a spatial constraint.	ogc:spatialOps or sos:ObjectID	Zero or one (Optional)
result	Provides a place to put in OGC filter expressions based on property values. This instructs the SOS to only return observations where the result matches this expression.	ogc:comparisonOps	Zero or one (Optional)
responseFormat	Specifies the desired resultFormat MIME content type for transport of the results (e.g. TML, O&M native format, or MPEG stream out-of-band). The supported output formats are listed in the selected offering capabilities.	Character String type (MIME content type)	One (mandatory)
resultModel	Specifies the QName of the root element of an O&M Observation or element in the appropriate substitution group. The resultModels supported by a SOS server are listed in the resultModel elements of the offerings in the service capabilities. If the requested resultModel is not supported by the SOS server, an exception message shall be returned.	QName	Zero or one (Optional)
responseMode	Specifies whether results are requested in-line, out-of-band, as an attachment, or if this is a request for an observation template that will be used for subsequent calls to GetResult. This is provided to allow the client to request the form of the response. The value of resultTemplate is used to retrieve an observation template that will later be used in calls to GetResult. The other options allow results to appear inline in a resultTag (inline), external to the observation element (out-of-band) or as a MIME attachment (attached).	inline out-of-band attached resultTemplate	Zero or one (Optional)

8.4.3   Response

The response to a GetObservation request is an O&M Observation, an element in the Observation substitution group, or an ObservationCollection. The SOS interface is optimized for access to observations and associated information. A detailed description of the response model and examples can be found in the Observations and Measurement encoding document[5].

8.4.4   Exceptions

When a SOS server encounters an error while performing a GetObservation operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

8.4.5   Examples

8.4.5.1   Example1 – Simple Request

In this example a client is requesting sensor data from a GID-3 chemical sensor that is advertised in the SOS capanbilities document. Here is the observation offering:

```xml
<ObservationOfferingList>
  <ObservationOffering gml:id="offering-3">
   <gml:name>urn:MyOrg:offering:3</gml:name>
   <gml:srsName>EPSG:4326</gml:srsName>
   <gml:boundedBy>
    <gml:Envelope srsName="EPSG:4326">
     <gml:lowerCorner>39.11111 -77.17695</gml:lowerCorner>
     <gml:upperCorner>40.11111 -77.17695</gml:upperCorner>
    </gml:Envelope>
   </gml:boundedBy>
   <time>
    <gml:TimePeriod>
     <gml:beginPosition>2005-10-18T19:54:13.000Z</gml:beginPosition>
     <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
   </time>
   <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
   <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12350"/>
   <observedProperty>
    <swe:CompositePhenomenon gml:id="AggregateChemicalPresence" dimension="2">
     <gml:name>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</gml:name>
     <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM"/>
     <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS"/>
    </swe:CompositePhenomenon>
   </observedProperty>
   <featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
   <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
   <resultModel>om:Observation</resultModel>
   <responseMode>inline</responseMode>
  </ObservationOffering>
```

Here is an example of a simple GetObservation request:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:om="http://www.opengis.net/om/1.0"
 service="SOS" version="1.0.0" srsName="EPSG:4326">
 <sos:offering>urn:MyOrg:offering:3</sos:offering>
 <sos:observedProperty>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</sos:observedProperty>
 <sos:responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</sos:responseFormat>
 <sos:resultModel>om:Observation</sos:resultModel>
 <sos:responseMode>inline</sos:responseMode>
</sos:GetObservation>
```

An example of the response is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<om:ObservationCollection xmlns:gml="http://www.opengis.net/gml"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
 xsi:schemaLocation="http://www.opengis.net/om/1.0  http://schemas.opengis.net/om/1.0.0/om.xsd"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <om:member>
  <om:Observation>
   <gml:name>urn:MyOrg:observation:1234abcde</gml:name>
   <om:samplingTime>
    <gml:TimePeriod>
     <gml:beginPosition> 2005-08-05T12:21:13Z </gml:beginPosition>
     <gml:endPosition> 2005-08-05T12:23:59Z </gml:endPosition>
    </gml:TimePeriod>
   </om:samplingTime>
   <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
   <om:observedProperty xlink:href="urn:ogc:def:property:MyOrg:AggregateChemicalPresence"/>
   <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
   <om:result>
    <swe:DataArray gml:id="ChemicalRecords">
     <swe:elementCount>
      <swe:Count>
       <swe:value>5</swe:value>
      </swe:Count>
     </swe:elementCount>
     <swe:elementType name="Components">
      <swe:SimpleDataRecord gml:id="DataDefinition">
       <swe:field name="time">
        <swe:Time definition="urn:ogc:property:time:iso8601"/>
       </swe:field>
       <swe:field name="longitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:longitude">
         <swe:uom code="deg"/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="latitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:latitude">
         <swe:uom code="deg"/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="DPM">
        <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
         <swe:uom code="ppm"/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="MS">
        <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
         <swe:uom code="ppm"/>
        </swe:Quantity>
       </swe:field>
      </swe:SimpleDataRecord>
```

```
      </swe:elementType>
      <swe:encoding>
       <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
      </swe:encoding>
      <swe:values> 2005-08-05T12:21:13Z,-77.8912,38.512,20,0@@
        2005-08-05T12:22:08Z,-77.8912,38.512,30,0@@ 2005-08-05T12:22:54Z,-77.8912,38.512,0,0@@
        2005-08-05T12:23:03Z,-77.8912,38.512,10,10@@ 2005-08-05T12:23:59Z,-77.8912,38.512,10,0
      </swe:values>
     </swe:DataArray>
    </om:result>
   </om:Observation>
 </om:member>
 <om:member>
  <om:Observation>
   <gml:name>urn:MyOrg:observation:1234abcdf</gml:name>
   <om:samplingTime>
    <gml:TimePeriod>
     <gml:beginPosition> 2005-08-05T12:23:03Z </gml:beginPosition>
     <gml:endPosition> 2005-08-05T12:23:59Z </gml:endPosition>
    </gml:TimePeriod>
   </om:samplingTime>
   <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12350"/>
   <om:observedProperty xlink:href="urn:ogc:def:property:MyOrg:AggregateChemicalPresence"/>
   <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
   <om:result>
    <swe:DataArray gml:id="ChemValues2">
     <swe:elementCount>
      <swe:Count>
       <swe:value>2</swe:value>
      </swe:Count>
     </swe:elementCount>
     <swe:elementType name="Components" xlink:href="#DataDefinition"/>
     <swe:encoding>
      <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
     </swe:encoding>
     <swe:values> 2005-08-05T12:23:03Z,-77.8912,38.512,10,10@@
       2005-08-05T12:23:59Z,-77.8912,38.512,10,0 </swe:values>
    </swe:DataArray>
   </om:result>
  </om:Observation>
 </om:member>
</om:ObservationCollection>
```

There are two observation instances returned in the example. The two observations are each for different sensors and the first sensor has five observations and the seconds sensor has two observations that satisfy the request.

8.4.5.2   Example2 – Request With Time Period

In this example the client is requesting all sensor data between 5:18pm and 58 seconds and 9:18pm and 59 seconds on November 5, 2006.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
 xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengis.net/ows/1.1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <offering>urn:MyOrg:SCAN_DATA</offering>
 <eventTime>
  <ogc:TM_During>
   <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
```

```
  <gml:TimePeriod>
    <gml:beginPosition>2006-11-05T17:18:58.000-06:00</gml:beginPosition>
    <gml:endPosition>2006-11-05T21:18:59.000-06:00</gml:endPosition>
  </gml:TimePeriod>
 </ogc:TM_During>
</eventTime>
<observedProperty>urn:ogc:def:property:MyOrg:BAND1</observedProperty>
<observedProperty>urn:ogc:def:property:MyOrg:BAND2</observedProperty>
<responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
</GetObservation>
```

8.4.5.3   Example3 – Request With Bounding Box

In this example the client is requesting all data within a bounding box.

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:om="http://www.opengis.net/om/1.0"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0" srsName="EPSG:4326">
 <sos:offering>urn:MyOrg:offering:3</sos:offering>
 <sos:observedProperty>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</sos:observedProperty>
 <sos:featureOfInterest>
  <ogc:BBOX>
    <ogc:PropertyName>gml:location</ogc:PropertyName>
    <gml:Envelope>
     <gml:lowerCorner>38.11 -78.6</gml:lowerCorner>
     <gml:upperCorner>38.14 -78.4</gml:upperCorner>
    </gml:Envelope>
  </ogc:BBOX>
 </sos:featureOfInterest>
 <sos:responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</sos:responseFormat>
 <sos:resultModel>om:Observation</sos:resultModel>
 <sos:responseMode>inline</sos:responseMode>
</sos:GetObservation>
```

9   Transaction Operations Profile (optional)

9.1   RegisterSensor (optional)

9.1.1   Introduction

The **RegisterSensor** operation allows the client to register a new sensor system with the SOS as part of the transactional profile. Sensor observations can only be inserted for sensors that have first been registered with the SOS. **RegisterSensor** is mandatory for the transactional profile.

9.1.2   Request

A RegisterSensor request includes a sensor system description, such as a SensorML or TML document, and an O&M Observation instance that is a template for the observations that will be published for this sensor using the InsertObservaton operation described below.

Each RegisterSensor operation has mandatory attributes of service and version. The version attribute must correspond to the specific service interface version negotiated between the service and client during the service binding process. The required service attribute explicitly specifies the service type. A fixed value of SOS must be used for the SOS interface.
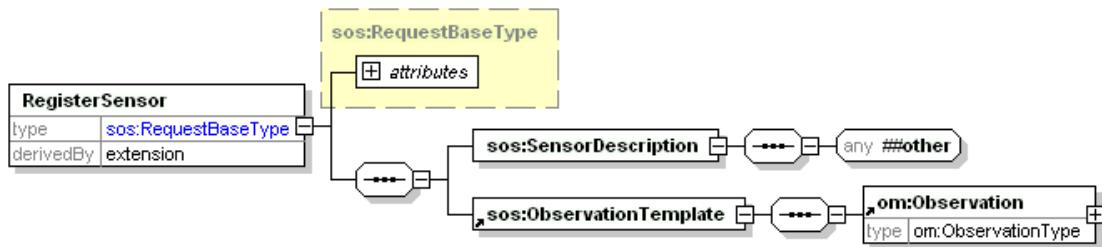
Figure 9-1 Parameters of a RegisterSensor Request

Table 5 Parameters of RegisterSensor Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
SensorDescription	This is generally a Sensor or System element from either SensorML or TML which is the detailed description of this sensor or system. The content is open to accommodate changes to TML and SensorML and to support other sensor description languages.	XML document	One (mandatory)
ObservationTemplate	A template for observation instances that will be inserted for this sensor or system.	om:Observation	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

The SOS advertises which sensor system description formats it supports in the operations metadata section of the capabilities documents. The following example shows an SOS that supports both TML and SensorML formats.

```
<ows:Operation name="RegisterSensor">
   <ows:DCP>
    <ows:HTTP>
     <ows:Post xlink:href="http://www.myorg.org/sos/RegisterSensor"/>
    </ows:HTTP>
   </ows:DCP>
   <ows:Constraint name="'SupportedSensorDescription">
    <ows:AllowedValues>
```

```
      <ows:Value>sml:System</ows:Value>
      <ows:Value>tml:system</ows:Value>
    </ows:AllowedValues>
    <ows:Meaning ows:reference="urn:ogc:def:swe:SupportedSensorDescription">The service will
      only accept sensor descriptions that comply with the listed ones.</ows:Meaning>
  </ows:Constraint>
 </ows:Operation>
```

9.1.3   Response

The response to a RegisterSensor request contains an AssignedSensorId which is the identifier assigned by the SOS to designate the new sensor. This identifier is of type anyURI and can be used in a subsequent DescribeSensor call. The identifier must be either a URN or a URL that resolves to a procedure that is advertised in an offering in the SOS capabilities response.

9.1.4   Exceptions

When a SOS server encounters an error while performing a RegisterSensor operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

9.1.5   Examples

See Annex B for an example of the RegisterSensor request and response.

9.2   InsertObservation (optional)

9.2.1   Introduction

The **InsertObservation** operation allows the client to insert new observations for a sensor system. This is a request to an SOS to perform the Insert operation. The request includes the sensor ID that was obtained from the RegisterSensor operation. The observation must be encoded in XML following the O&M specification. InsertObservation is mandatory for the transactional profile.

9.2.2   Request

An InsertObservation request includes the AssignedSensorId that was returned from the RegisterSensor operation when the sensor that made this observation was registered. It also includes an O&M Observation that follows the template that was provided when the sensor was registered.

Each InsertObservation operation has mandatory attributes of service and version. The version attribute must correspond to the specific service interface version negotiated between the service and client during the service binding process. The required service attribute explicitly specifies the service type. A fixed value of SOS must be used for the SOS interface.
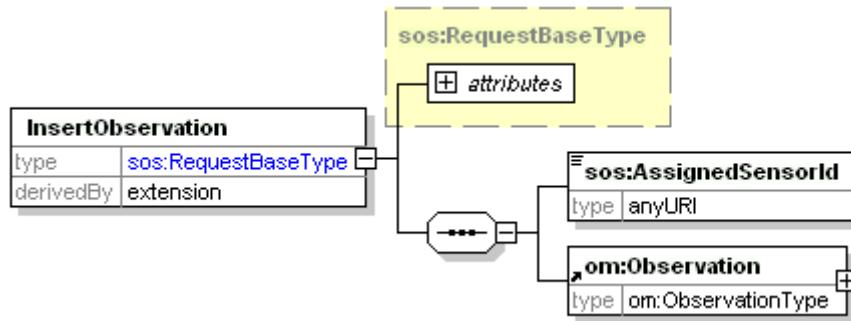
Figure 9-2 Parameters of an InsertObservation Request

Table 6 Parameters of insertObservation Request

Name [a]	Definition	Data type and values	Multiplicity and use
AssignedSensorId	Specifies the identifier for the sensor that made this observation. This identifier is obtained by the RegisterSensor operation.	anyURI	One (mandatory)
Observation	The new observation for insertion.	om:Observation	One (mandatory)
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	The supported version of SOS	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

9.2.3   Response

The response to an InsertObservation request is the assigned Observation identifier. This identifier may be used to obtain this observation from the SOS using the GetObservationById operation.

9.2.4   Exceptions

When a SOS server encounters an error while performing an InsertObservation operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

9.2.5   Examples

9.2.5.1   Example1

Here is an example of an InsertObservation request:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<InsertObservation xmlns="http://www.opengis.net/sos/1.0" xmlns:om="http://www.opengis.net/om/1.0"
 xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:gml="http://www.opengis.net/gml"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```xml
xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
service="SOS" version="1.0.0">
<AssignedSensorId>urn:ogc:object:Sensor:MyOrg:12349</AssignedSensorId>
<om:Observation>
  <om:samplingTime>
    <gml:TimeInstant>
      <gml:timePosition>2005-08-05T12:21:13Z</gml:timePosition>
    </gml:TimeInstant>
  </om:samplingTime>
  <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
  <om:observedProperty>
    <swe:CompositePhenomenon gml:id="composite1" dimension="3">
      <gml:name codeSpace="urn:MyOrg">urn:ogc:def:property:MyOrg:AggregateChemicalPresence</gml:name>
      <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM"/>
      <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS"/>
      <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA"/>
    </swe:CompositePhenomenon>
  </om:observedProperty>
  <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
  <om:result>
    <swe:SimpleDataRecord>
      <swe:field name="time">
        <swe:Time definition="urn:ogc:property:time:iso8601">
          <swe:value>2005-08-05T12:21:13Z</swe:value>
        </swe:Time>
      </swe:field>
      <swe:field name="longitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:longitude">
          <swe:uom code="deg"/>
          <swe:value>-77.8912</swe:value>
        </swe:Quantity>
      </swe:field>
      <swe:field name="latitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:latitude">
          <swe:uom code="deg"/>
          <swe:value>38.512</swe:value>
        </swe:Quantity>
      </swe:field>
      <swe:field name="DPM">
        <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
          <swe:uom code="ppm"/>
          <swe:value>20.1</swe:value>
        </swe:Quantity>
      </swe:field>
      <swe:field name="MS">
        <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
          <swe:uom code="ppm"/>
          <swe:value>0.0</swe:value>
        </swe:Quantity>
      </swe:field>
    </swe:SimpleDataRecord>
  </om:result>
</om:Observation>
</InsertObservation>
```

Here is an example of an InsertObservation response:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<InsertObservationResponse xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <AssignedObservationId>urn:MyOrg:Observation:5678</AssignedObservationId>
</InsertObservationResponse>
```

10 Enhanced Operations Profile (optional)

10.1 GetObservationById (optional)

10.1.1 Introduction

The GetObservationById operation is designed to return an observation based on an identifier. This identifier must have been obtained by the client prior to invoking the operation. The id might have come from a link in another XML document or from a previous call to InsertObservation.

10.1.2 Request

A <GetObservationById> message contains the id of the rrequested observation as well as optional parameters governing the form of the returned observation. Each GetObservationById element has mandatory attributes of service and version. The mandatory version element attribute must correspond to the specific service interface version negotiated between the service and client during the service binding process. The required service attribute explicitly must be "SOS".
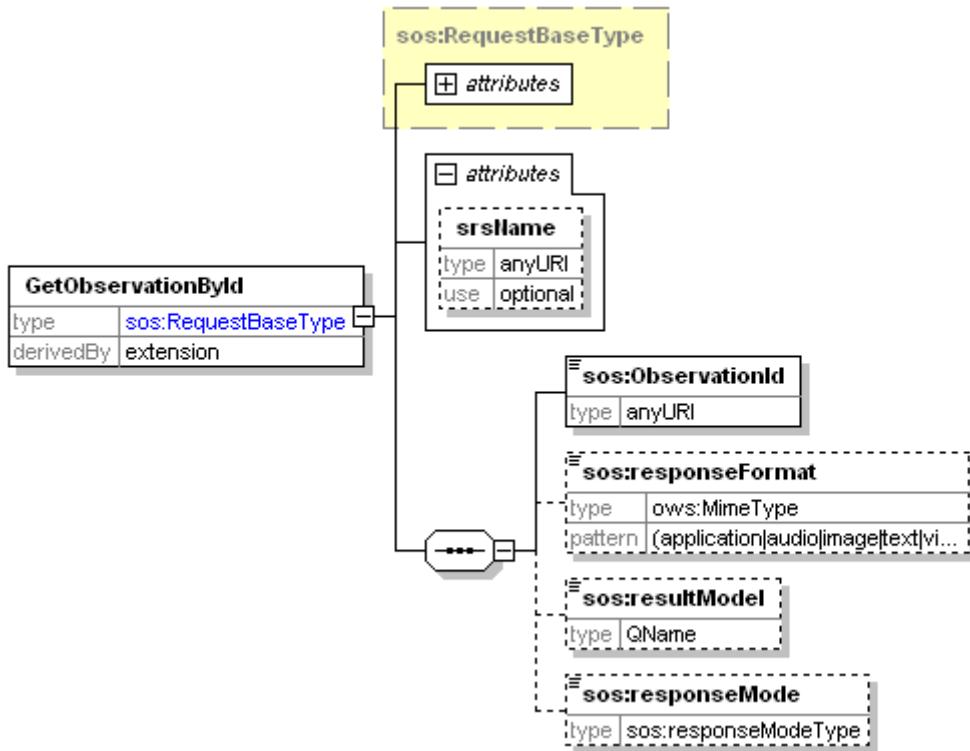
Figure 10-1 GetObservationById Request

Table 7 Parameters of GetObservation Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Verson of SOS supported.	One (mandatory)
srsName	Defines the spatial reference system that should be used for any geometries that are returned in the response. This must be one of the advertised values in the offering specified in gml:srsName elements.	anyURI	One (optional)
ObservationId	The Id of the requested observation.	anyURI	One (mandatory)
responseFormat	Specifies the desired resultFormat MIME content type for transport of the results (e.g. TML, O&M native format, or MPEG stream out-of-band). The supported output formats are listed in the selected offering capabilities.	Character String type (MIME content type)	One (mandatory)
resultModel	Specifies the QName of the root element of an O&M Observation or element in the appropriate substitution group. The resultModels supported by a SOS server are listed in the contents section of the service metadata, identified as QName values. If the requested resultModel is not supported by the SOS server, an exception message shall be returned.	QName	Zero or one (Optional)
responseMode	Specifies whether results are requested in-line, out-of-band, as an attachment, or if this is a request for an observation template that will be used for subsequent calls to GetResult. This is provided to allow the client to request the form of the response. The value of resultTemplate is used to retrieve an observation template that will later be used in calls to GetResult. The other options allow results to appear inline in a resultTag (inline), external to the observation element (out-of-band) or as a MIME attachment (attached).	inline out-of-band attached resultTemplate	Zero or one (Optional)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.1.3 Response

The response to a GetObservationById request is an element in the om:Observation substitution group or an ObservationCollection. The exact element returned depends on the resultModel parameter.

10.1.4 Exceptions

When a SOS server encounters an error while performing a GetObservationById operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.1.5 Examples

10.1.5.1 Example1

Example request:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservationById xmlns:sos="http://www.opengis.net/sos/1.0"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0" srsName="EPSG:4326">
 <sos:ObservationId>urn:MyOrg:Observation:5678</sos:ObservationId>
 <sos:resultModel>om:Observation</sos:resultModel>
</sos:GetObservationById>
```

Example response:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<om:Observation xmlns:sos="http://www.opengis.net/sos/1.0"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
 xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/om/1.0  http://schemas.opengis.net/om/1.0.0/om.xsd">
 <gml:name>urn:MyOrg:Observation:5678</gml:name>
 <om:samplingTime>
  <gml:TimeInstant>
   <gml:timePosition>2005-08-05T12:21:13Z</gml:timePosition>
  </gml:TimeInstant>
 </om:samplingTime>
 <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
 <om:observedProperty>
  <swe:CompositePhenomenon gml:id="composite1" dimension="3">
   <gml:name codeSpace="urn:MyOrg">urn:ogc:def:property:MyOrg:AggregateChemicalPresence</gml:name>
   <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM"/>
   <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS"/>
   <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA"/>
  </swe:CompositePhenomenon>
 </om:observedProperty>
 <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
 <om:result>
  <swe:SimpleDataRecord>
   <swe:field name="time">
    <swe:Time definition="urn:ogc:def:property:OGC:OriginTime">
     <swe:value>2005-08-05T12:21:13Z</swe:value>
    </swe:Time>
   </swe:field>
   <swe:field name="latitude">
    <swe:Quantity definition="urn:ogc:def:property:OGC:Latitude:wgs84">
     <swe:uom code="deg"/>
     <swe:value>-77.8912</swe:value>
    </swe:Quantity>
   </swe:field>
   <swe:field name="longitude">
    <swe:Quantity definition="urn:ogc:def:property:OGC:Longitude:wgs84">
     <swe:uom code="deg"/>
     <swe:value>38.512</swe:value>
    </swe:Quantity>
   </swe:field>
   <swe:field name="altitude">
    <swe:Quantity definition="urn:ogc:def:property:OGC:Altitude:wgs84">
     <swe:uom code="m"/>
     <swe:value>20</swe:value>
    </swe:Quantity>
```

```
    </swe:field>
    <swe:field name="DPM">
     <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
      <swe:uom code="ppm"/>
      <swe:value>0</swe:value>
     </swe:Quantity>
    </swe:field>
    <swe:field name="MS">
     <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
      <swe:uom code="ppm"/>
      <swe:value>12</swe:value>
     </swe:Quantity>
    </swe:field>
    <swe:field name="GA">
     <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA">
      <swe:uom code="ppm"/>
      <swe:value>8</swe:value>
     </swe:Quantity>
    </swe:field>
   </swe:SimpleDataRecord>
  </om:result>
</om:Observation>
```

10.2   GetResult (optional)

10.2.1 Introduction

The purpose of the GetResult operation is to allow a client to repeatedly obtain sensor data from the same set of sensors without having to send and receive requests and responses that largely contain the same data except for a new timestamp. A common use case is for a client to repeatedly request current sensor data from one or more sensors on a recurring basis. This operation supports that use case and allows it to occur using much less bandwidth than would be necessary for a full GetObservation call. The motivation for including the operation is to support a data center requesting data from a node that talks directly to sensors over a low bandwidth connection such as a 3G wireless link.

The GetResult operation relies on the creation of an O&M template from a previous call to GetObservation. The identifier of the template is used for subsequent GetResult calls instead of sending a duplicate GetObservation XML document. The response contains only the result portion of the O&M Observation because the other components are included by reference in the template.

Here is a typical sequence of even that might occur with the GetResult operation:

1. Client sends a GetObservation request with responseMode="resultTemplate" indicating that he wants an observation template for later calls to GetResult

2. SOS responds with a collection of Observation templates. There is one observation template per sensor that matches the query criteria. Each template has a GML name that is unique to the SOS server. Each template has a valid time period which acts like a lease. The template is no longer valid after the expiration and subsequent GetResult requests fail after the expiration.

3. Client sends a GetResult request with no time parameter.

4. SOS Server responds with result(s) including all observations available since the begin time of the lease.

5. Client sends a GetResult request with an "after" time representing the latest time of an observation from the previous request. This allows the client to filter out duplicates and means the server does not need to maintain state for each GetResult request.

6. SOS Server responds with result(s) including all observations after the time specified.

7. Client sends request...

8. SOS server responds...

etc. etc

10.2.2 Request
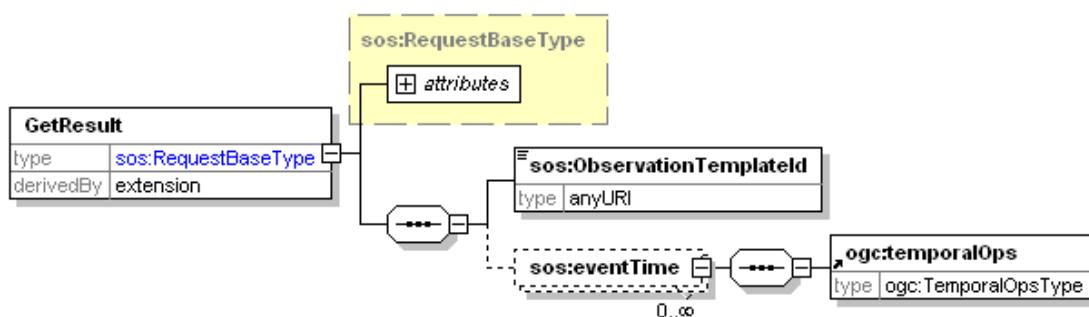
Figure 10-2 Parameters of GetResult Request

Table 8. Parameters of GetResult Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
ObservationTemplateId	Specifies the Observation template whose result is desired. This must match the gml:name of an Observation from a template that resulted from a previous call to GetObservation.	anyURI	One (mandatory)
eventTime	Time-based filter expression.	ogc:temporalOps	Optional
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.2.3 Response

The response to a GetResult request is a GetResultResponse document. The response contains only a result element with an RS attribute. The RS attribute is a URI that

reference the O&M template that provides the context for the in-line result value. The template is the same as what was returned from the call to GetObservation that preceded the call to GetResult (see explanation above). The result is the contents of the result that would be inserted in the referenced template to create a complete O&M Observation document.

10.2.4 Exceptions

When a SOS server encounters an error while performing a GetResult operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.2.5 Examples

An example of the GetResult call sequence is included in Annex B.

10.3   GetFeatureOfInterest (optional)

10.3.1 Introduction

GetFeatureOfInterest returns a featureOfInterest that was advertised in one of the observation offerings of the SOS capabilities document. This could be a Station for in-situ sensors, for example.

10.3.2 Request
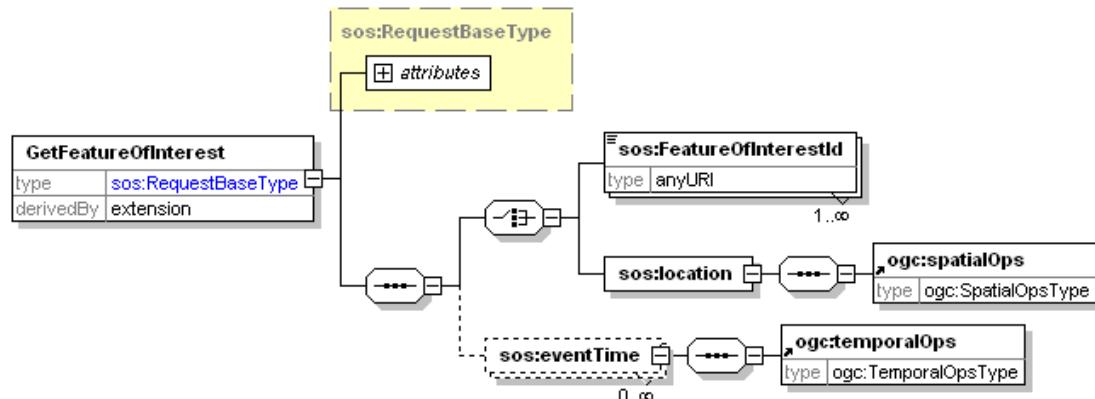
Figure 10-3 Parameters for a GetFeatureOfInterest Request

Table 9. Parameters of GetFeatureOfInterest Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
FeatureOfInterestId [b]	Specifies the identifier of the feature of interest being returned, for which detailed information is requested. These identifiers must be listed in the Contents section of the service metadata (GetCapabilities) document.	anyURI	Choice 1 - One (mandatory)
location	This argument allows the request to be made based on a spatial filter expression rather than by using an actual feature ID.	ogc:spatialOpsType	Choice 2 – One (mandatory)
eventTime	Specifies the time for which the feature of interest is to be queried. This uses a modified version of filter.xsd and allows a client to request targets from a specific instance, multiple instances or periods of time in the past, present and future. This is useful for dynamic sensors for which the properties of the feature of interest are time-dependent. Multiple time parameters may be indicated so that the client may request details of the observation target at multiple times. The supported range is listed in the contents section of the service metadata.	ogc:temporalOps	One (optional)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			
b The schema presents a choice between requesting based on Id or location			

10.3.3 Response

The response to a GetFeatureOfInterest request is a GML feature instance. This may be a domain-specific feature type, such as Tract, Paddock, Aquifer, Road, or may be a "Sampling Feature" such as Station, Specimen, Profile, Scene (see OGC 07-002).

10.3.4 Exceptions

When a SOS server encounters an error while performing a GetFeatureOfInterest operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.3.5 Examples

10.3.5.1 Example1

Here is an example of a capabilities document with a reference to a featureOfInterest that happens to be a Station.

```
<ObservationOffering gml:id="offering-1">
    <gml:name>urn:MyOrg:offering:1</gml:name>
…
    <featureOfInterest xlink:href="urn:MyOrg:feature:station1"/>
…
</ObservationOffering>
```

The client wants to retrieve the feature instance for the station to see where it is located. Here is what the request might look like. Note that the FeatureOfInterestId matches the xlink:href of the featureOfInterest from the capabilities document.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetFeatureOfInterest xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <FeatureOfInterestId>urn:MyOrg:feature:station1</FeatureOfInterestId>
</GetFeatureOfInterest>
```

The response might look like this indicating the location of the station and some additional metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
<sa:SamplingPoint gml:id="st1" xmlns:sa="http://www.opengis.net/sampling/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:gml="http://www.opengis.net/gml"
 xsi:schemaLocation="http://www.opengis.net/sampling/1.0
http://schemas.opengis.net/sampling/1.0.0/sampling.xsd">
 <gml:name>urn:MyOrg:feature:station1</gml:name>
 <sa:sampledFeature xlink:href="http://MyOrg.org/features/atmosphere/abc345"/>
 <sa:position>
  <gml:Point gml:id="st1p">
   <gml:pos srsName="urn:ogc:def:crs:EPSG:6.8:4283">-30.702 134.199</gml:pos>
  </gml:Point>
 </sa:position>
</sa:SamplingPoint>
```

10.4   GetFeatureOfInterestTime (optional)

10.4.1 Introduction

GetFeatureOfInterestTime returns the time periods for which the SOS will return data for a given advertised feature of interest.

10.4.2 Request
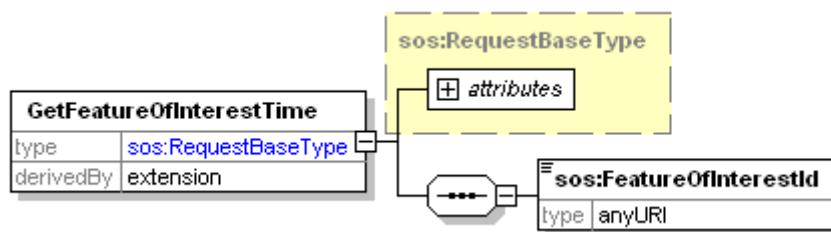
Figure 10-4 Parameters for a GetFeatureOfInterestTime Request

Table 10. Parameters of GetFeatureOfInterestTime Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
FeatureOdInterestId	Request that a feature be selected by ID.	anyURI	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.4.3 Response

The response to a GetFeatureOfInterestTime request is a GML time primitive which lists one or more time periods for which observations from that feature of interest are available.

10.4.4 Exceptions

When a SOS server encounters an error while performing a GetFeatureOfInterestTime operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.4.5 Examples

10.4.5.1 Example1

Here is an example of a capabilities document with a reference to a featureOfInterest that happens to be a Station.

```
<ObservationOffering gml:id="offering-1">
    <gml:name>urn:MyOrg:offering:1</gml:name>
...
    <featureOfInterest xlink:href="urn:MyOrg:feature:station1"/>
...
</ObservationOffering>
```

Here is a GetFeatureOfInterestTime request that seeks the times for which this station provides observations.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetFeatureOfInterestTime xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <FeatureOfInterestId>urn:MyOrg:feature:station1</FeatureOfInterestId>
</GetFeatureOfInterestTime>
```

Here is what the response might look like. In this example the SOS is reporting that data from station 1 is available from 2005-10-25T14:45:00.000Z to 2005-11-01T01:25:00.000Z.

```
<gml:TimePeriod xmlns:gml="http://www.opengis.net/gml">
 <gml:beginPosition>2005-10-25T14:45:00.000Z</gml:beginPosition>
 <gml:endPosition>2005-11-01T01:25:00.000Z</gml:endPosition>
</gml:TimePeriod>
```

10.5   DescribeFeatureType (optional)

10.5.1 Introduction

DescribeFeatureType returns the XML schema for the specified GML feature advertised in GetCapabilities. This may be used to obtain a description of the type of an observation feature-of-interest.

10.5.2 Request
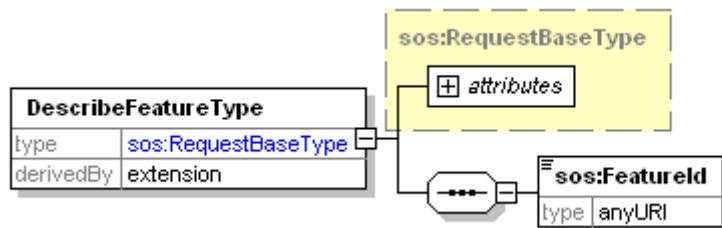
Figure 10-5 Parameters for DescribeFeatureType Request

Table 11. Parameters of DescribeFeatureType Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
FeatureId	The xlink:href of the feature of interest from the capabilities document.	anyURI	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.5.3 Response

The response to a DescribeFeatureType request is the XML Schema that defines the feature type for the specified feature.

10.5.4 Exceptions

When a SOS server encounters an error while performing a DescribeFeatureType operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.5.5 Examples

10.5.5.1 Example1

Here is an example of a DescribeFeatureType request for the schema of the feature called station 1 used in the previous examples.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DescribeFeatureType xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <FeatureId>urn:MyOrg:feature:station1</FeatureId>
</DescribeFeatureType>
```

The response would be the schema for Station which might look like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:gml="http://www.opengis.net/gml" xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:om="http://www.opengis.net/om/1.0"
xmlns="http://www.w3.org/2001/XMLSchema"
     xmlns:sa="http://www.opengis.net/sampling/1.0" targetNamespace="http://www.opengis.net/sampling/1.0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0.0">

. . .

    <complexType name="StationType">
        <annotation>
            <documentation>A "Station" is an identified position (0-D geospatial feature).
        It may be revisited for various purposes, in particular to retrieve multiple specimens or make repeated or
complementary observations.
        The position property of the station provides the value of the position property of observations an
specimens associated with the station.
        By using the Station feature, this position information and any metadata associated with it may be encoded
in one place, i.e. normalised, and then re-used _by reference_ on other feature instances associated with
it.</documentation>
        </annotation>
        <complexContent>
            <extension base="sa:SamplingFeatureType">
                <sequence>
                    <element name="position" type="gml:PointPropertyType"/>
                    <element name="elevation" type="gml:DirectPositionType" minOccurs="0">
                        <annotation>
                            <documentation>Use the srsName attribute to record the elevation
datum</documentation>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!-- ................. -->
    <element name="Station" type="sa:StationType" substitutionGroup="sa:SamplingFeature"/>

. . .

</schema>
```

10.6   DescribeObservationType (optional)

10.6.1 Intoduction

The DescribeObservationType operation returns the XML schema that describes the Observation type that is returned for a particular phenomenon. This allows the SOS to list the set of Observation types that it can deliver (by default the value is "om:Observation"). These are specialized observation types, in which one or other of the standard properties are specialised or restricted, or additional properties added. For example om:Measurement is a specialization in which the result is type="gml:MeasureType". It might be an observation type defined in a specific application domain (e.g. GeoChemMeasurement, defined in XMML, WaterQualityMeasurement, defined by NSW-DIPNR, etc.).

10.6.2 Request
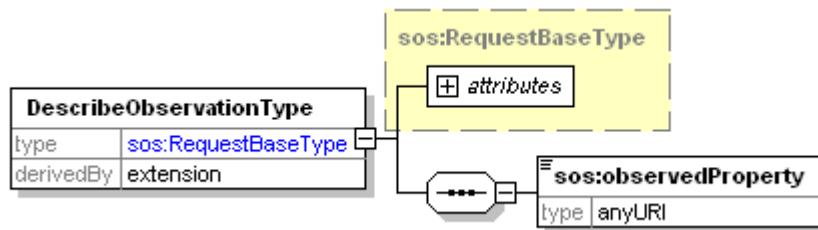
Figure 10-6 Parameters for a DescribeObservationType Request

Table 12 Parameters of DescribeObservationType Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
observedProperty	Reference to the observedProperty identified in the capablities docuement.	anyURI	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.6.3 Response

The response to a DescribeObservationType request is the XML Schema that defines the specialized observation type returned for the given observedProperty.

10.6.4 Exceptions

When a SOS server encounters an error while performing a DescribeObservationType operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.6.5 Examples

10.6.5.1 Example1

Here is an example of a DescribeObservationType request for the phenomenon of wind speed.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DescribeObservationType xmlns="http://www.opengis.net/sos/1.0"
 xmlns:ows="http://www.opengis.net/ows/1.1" service="SOS" version="1.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0
 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <observedProperty>urn:ogc:def:property:MyOrg:WindSpeed</observedProperty>
</DescribeObservationType>
```

The response indicates that the SOS will use the Measurement observation type to report this phenomenon.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:meta="http://www.seegrid.csiro.au/xml/metaLite"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
targetNamespace="http://www.opengis.net/om/1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">

. . .

    <complexType name="MeasurementType">
        <complexContent>
            <extension base="om:ObservationType">
                <sequence>
                    <element name="result" type="gml:MeasureType" nillable="true"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Measurement" type="om:MeasurementType" substitutionGroup="om:Observation">
        <annotation>
            <documentation>Measurement event</documentation>
        </annotation>
    </element>
</schema>
```

10.7   DescribeResultModel (optional)

10.7.1 Introduction

DescribeResultModel returns the schema for the result element that will be returned when the client asks for the given result model by the given ResultName. The "generic" om:Observation has result type ="xs:anyType", so it is not until an instance is received that the client discovers what the model for the result is, either by inspection of the value

of an xsi:type attribute on the <result> element, or merely through the appearance of a sub-element from a namespace declared for the document or local context, with a schemaLocation provided. In order to be able to prepare to use this, the result of a **DescribeResultModel** operation advises the client what the result will contain. A similar argument applies to the other observation types that have soft-typed results, i.e. ComplexObservation, ComplexMeasurement. The qualified name of the result element can be obtained from the schema that is returned by DescribeObservationType.

10.7.2 Request
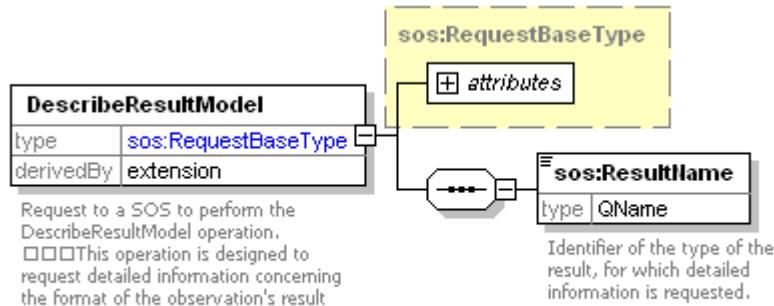
Figure 10-7 Parameters for the DescribeResultModel Request

Table 13 Parameters of DescribeResultModel Request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	SOS	One (mandatory)
version	Specification version for operation	Supported version of SOS	One (mandatory)
ResultName	The ResultName parameter specifies the qualified name of the <result> element in an Observation for which the DescribeResultModel operation is performed.	QName	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].			

10.7.3 Response

The response to a DescribeResultModel request is the XML Schema that defines the specialized result element type returned for the given qualified name.

10.7.4 Exceptions

When a SOS server encounters an error while performing a DescribeResultModel operation, it shall return an exception report message as specified in clause 8 of [OGC 06-121r3].

10.7.5 Examples

10.7.5.1 Example1

Here is an example of a DescribeResultModel request for the qualified name gml:MeasureType that was obtained from the call to DescribeObservationType in the previous example.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DescribeResultModel xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
 xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:om="http://www.opengis.net/om/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0   http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <ResultName>gml:MeasureType</ResultName>
</DescribeResultModel>
```

The response is the XML schema for gml:MeasureType.

```xml
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.1.1">

. . .

<complexType name="MeasureType">
        <annotation>
            <documentation>Number with a scale.
    The value of uom (Units Of Measure) attribute is a reference to a Reference System for the amount, either a
ratio or position scale. </documentation>
        </annotation>
        <simpleContent>
            <extension base="double">
                <attribute name="uom" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

. . .

</schema>
```

Annex A
(normative)

SOS Schema

In addition to this document, this specification includes normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. These XML Schema Documents will also be posted online at the URL http://schemas.opengeospatial.net/sos/1.0.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative.

A.1 *GetCapabilities* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:sos="http://www.opengis.net/sos/1.0"
targetNamespace="http://www.opengis.net/sos/1.0" elementFormDefault="qualified" version="1.0.0"
xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema encodes the SOS GetCapabilities operation request and
response.</description>
  </documentation>
 </annotation>
 <!-- ==============================================================
     and imports
     ============================================================== -->
 <include schemaLocation="sosContents.xsd"/>
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <import namespace="http://www.opengis.net/ogc" schemaLocation="./ogc4sos.xsd"/>
 <!-- ==============================================================
     and types
     ============================================================== -->
 <element name="GetCapabilities">
  <annotation>
   <documentation>Request to a SOS to perform the GetCapabilities operation. This operation allows a client to
retrieve service metadata (capabilities XML) providing metadata for the specific SOS server. In this XML encoding,
no "request" parameter is included, since the element name specifies the specific operation. </documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="ows:GetCapabilitiesType">
     <sequence/>
     <attribute name="service" type="ows:ServiceType" use="required" fixed="SOS"/>
    </extension>
   </complexContent>
  </complexType>
 </element>
 <!-- ============================================================ -->
 <element name="Capabilities">
  <annotation>
   <documentation>XML encoded SOS GetCapabilities operation response. This document provides clients with
service metadata about a specific service instance, including metadata about the tightly-coupled data served. If the
server does not implement the updateSequence parameter, the server shall always return the complete
Capabilities document, without the updateSequence parameter. When the server implements the updateSequence
```

parameter and the GetCapabilities operation request included the updateSequence parameter with the current value, the server shall return this element with only the "version" and "updateSequence" attributes. Otherwise, all optional elements shall be included or not depending on the actual value of the Sections parameter in the GetCapabilities operation request. </documentation>
```xml
    </annotation>
    <complexType>
     <complexContent>
      <extension base="ows:CapabilitiesBaseType">
       <sequence>
        <element ref="sos:Filter_Capabilities" minOccurs="0"/>
        <element ref="sos:Contents" minOccurs="0"/>
       </sequence>
      </extension>
     </complexContent>
    </complexType>
   </element>
   <!-- added by Mike Botts and Johannes Echterhoff to support temporal filter in SOS Capabailities 2007-10-18 -->
   <!-- temporary fix until temporal ops supported in filter -->
   <element name="Filter_Capabilities">
    <complexType>
     <sequence>
      <element ref="ogc:Spatial_Capabilities"/>
      <element ref="ogc:Temporal_Capabilities"/>
      <element ref="ogc:Scalar_Capabilities"/>
      <element ref="ogc:Id_Capabilities"/>
     </sequence>
    </complexType>
   </element>
  </schema>
```

A.2 *DescribeSensor* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema defines the DescribeSensor request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ===============================================================
     and imports
     =============================================================== -->
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ===============================================================

     =============================================================== -->
 <element name="DescribeSensor">
  <annotation>
   <documentation>Request to a SOS to perform the DescribeSensor operation. This operation is designed to
request detailed sensor metadata.    </documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="procedure" type="anyURI">
       <annotation>
        <documentation>Identifier of the sensor, for which detailed metadata is requested.</documentation>
       </annotation>
      </element>
```

```
      </sequence>
      <attribute name="outputFormat" type="ows:MimeType" use="required">
        <annotation>
          <documentation>Identifier of the output format to be used for the requested data. The outputFormats
supported by a SOS server are listed in the operations metadata section of the service metadata (capabilities
XML). If this attribute is omitted, the outputFormat should be tex/xml;subtype="sensorML/1.0.0". If the requested
outputFormat is not supported by the SOS server, an exception message shall be returned.
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
 </complexType>
</element>
<!-- ================================================================
```

```
     ================================================================ -->
</schema>
```

A.3 *GetObservation* Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" xmlns:om="http://www.opengis.net/om/1.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:dc="http://purl.org/dc/elements/1.1/" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
    <dc:description>This XML Schema defines the GetObservation request XML elements and
types.</dc:description>
  </documentation>
 </annotation>
 <!-- ================================================================
     and imports
     ================================================================ -->
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <import namespace="http://www.opengis.net/ogc" schemaLocation="ogc4sos.xsd"/>
 <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ================================================================
```

```
     ================================================================ -->
 <element name="GetObservation">
  <annotation>
    <documentation>Request to a SOS to perform the GetObservation operation. This operation is designed to
request sensor data from live sensors as well as sensor data archives.</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence minOccurs="0">
      <element name="offering" type="anyURI">
        <annotation>
          <documentation>ID of an offering advertised in the capabilities.
                      following parameters are depending on the selected offering.
          </documentation>
        </annotation>
      </element>
      <element name="eventTime" minOccurs="0" maxOccurs="unbounded">
        <annotation>
```

```
      <documentation>Allows a client to request observations from a specific instant, multiple instances or
periods of time in the past, present and future. The supported range is listed in the selected offering capabilities.
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="ogc:temporalOps"/>
        </sequence>
      </complexType>
    </element>
    <element name="procedure" type="anyURI" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>Index of a particular sensor if offering procedure is a Sensor Array. Allows client to
request data from one or more sensors in the array. The size of the array should be specified in the selected
offering capabilities. This is to support scenarios with sensor grids (we don't want to have one offering for each
sensor in that case). Note that sensorML can describe Sensor Arrays too.
        </documentation>
      </annotation>
    </element>
    <element name="observedProperty" type="anyURI" maxOccurs="unbounded">
      <annotation>
        <documentation>ID of a phenomenon advertised in capabilities document.
                          possible phenomena are listed in the selected offering capabilities.
        </documentation>
      </annotation>
    </element>
    <element name="featureOfInterest" minOccurs="0">
      <annotation>
        <documentation>Specifies target feature for which observations are requested. Mostly a hepler for in-situ
sensors, since geo-location has to be done on the server side. The supported area should be listed in the selected
offering capabilities.
        </documentation>
      </annotation>
      <complexType>
        <choice>
          <element ref="ogc:spatialOps"/>
          <element name="ObjectID" type="anyURI" maxOccurs="unbounded">
            <annotation>
              <documentation>Unordered list of zero or more object identifiers. These identifiers are usually listed
in the Contents section of the service metadata (Capabilities) document. If no ObjectID value is included, and if
only one category of objects is allowed for this operation, the server shall return all objects of that category. NOTE:
Although retention of this ability is allowed by a specific OWS that uses this operation, such retention is
discouraged due to possible problems. Making this ability optional implementation by servers reduces
interoperability. Requiring implementation of this ability can require a server to return a huge response, when there
are a large number of items in that category. </documentation>
            </annotation>
          </element>
        </choice>
      </complexType>
    </element>
    <element name="result" minOccurs="0">
      <annotation>
        <documentation>Only report observations where the result matches this expression.
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="ogc:comparisonOps"/>
        </sequence>
      </complexType>
    </element>
    <element name="responseFormat" type="ows:MimeType">
      <annotation>
        <documentation>ID of the output format to be used for the requested data. The supported output formats
are listed in the selected offering capabilities.</documentation>
      </annotation>
    </element>
```

```xml
        <element name="resultModel" type="QName" minOccurs="0">
         <annotation>
           <documentation>Identifier of the result model to be used for the requested data. The resultModel values
supported by a SOS server are listed in the contents section of the service metadata, identified as QName values.
If the requested resultModel is not supported by the SOS server, an exception message shall be returned.
           </documentation>
         </annotation>
        </element>
        <element name="responseMode" type="sos:responseModeType" minOccurs="0">
         <annotation>
           <documentation>This element allows the client to request the form of the response.  The value of
resultTemplate is used to retrieve an observation template
                                will later be used in calls to GetResult.  The other options allow results to appear
inline in a resultTag (inline), external to the observation element (out-of-band)
                                as a MIME attachment (attached)</documentation>
         </annotation>
        </element>
       </sequence>
       <attribute name="srsName" type="anyURI" use="optional"/>
      </extension>
    </complexContent>
   </complexType>
 </element>
</schema>
```

A.4 *RegisterSensor* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
   <documentation>
     <description>This XML Schema defines the registerSenor request and response XML elements and
types.</description>
   </documentation>
 </annotation>
 <!-- ==============================================================

       and imports
       ============================================================== -->
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <import namespace="http://www.opengis.net/om/1.0"
schemaLocation="http://schemas.opengis.net/om/1.0.0/om.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ==============================================================


       ============================================================== -->
 <element name="RegisterSensor">
  <annotation>
   <documentation>Request to a SOS to perform the registerSensor operation. This operation is designed to
register new sensors at the SOS.</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="SensorDescription">
       <complexType>
        <sequence>
         <any namespace="##other" processContents="lax"/>
        </sequence>
       </complexType>
      </element>
```

57

```
        <element ref="sos:ObservationTemplate"/>
      </sequence>
    </extension>
   </complexContent>
  </complexType>
 </element>
 <element name="ObservationTemplate">
  <annotation>
   <documentation>A template of the observations that will be inserted into the SOS.</documentation>
  </annotation>
  <complexType>
   <sequence>
    <element ref="om:Observation"/>
   </sequence>
  </complexType>
 </element>
 <!-- =====================================================

      ============================================================-->
 <element name="RegisterSensorResponse">
  <annotation>
   <documentation>returns the Id that is used in the insert operation to link the sensor to an
Observation</documentation>
  </annotation>
  <complexType>
   <sequence>
    <element name="AssignedSensorId" type="anyURI"/>
   </sequence>
  </complexType>
 </element>
</schema>
```

A.5 *InsertObservation* Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema defines the insert request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ================================================================
      and imports
      ================================================================ -->
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <import namespace="http://www.opengis.net/om/1.0"
schemaLocation="http://schemas.opengis.net/om/1.0.0/om.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ================================================================

      ================================================================ -->
 <element name="InsertObservation">
  <annotation>
   <documentation>Request to a SOS to perform the Insert operation. This operation is designed to insert new
observations. The request is constraint by the following parameters: ID obtained by the registerSensor operation
(identifying the sensor and the observyationType, and the observation encoded as OM</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
```

```xml
  <sequence>
   <element name="AssignedSensorId" type="anyURI">
    <annotation>
     <documentation>The id obtained by the registerSensor operation.</documentation>
    </annotation>
   </element>
   <element ref="om:Observation">
    <annotation>
     <documentation>The observation to be inserted to the SOS.</documentation>
    </annotation>
   </element>
  </sequence>
 </extension>
</complexContent>
</complexType>
</element>
<!-- ===================================================

  ============================================================-->
<element name="InsertObservationResponse">
 <annotation>
  <documentation>returns the Id of the Observation assigend by the SOS</documentation>
 </annotation>
 <complexType>
  <sequence>
   <element name="AssignedObservationId" type="anyURI"/>
  </sequence>
 </complexType>
</element>
</schema>
```

A.6 *GetResult* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema defines the GetResult request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ================================================================
     and imports
     ================================================================ -->
<import namespace="http://www.opengis.net/ogc" schemaLocation="ogc4sos.xsd"/>
<include schemaLocation="sosCommon.xsd"/>
<!-- ================================================================

  ================================================================ -->
<element name="GetResult">
 <annotation>
  <documentation>request to a SOS to perform a GetResult operation. this operation is designed to request
sensor data from live sensors. Instead of retriveing the observations as a full OM document, you will get an simple
value and a link to the reference system</documentation>
 </annotation>
 <complexType>
  <complexContent>
   <extension base="sos:RequestBaseType">
    <sequence>
     <element name="ObservationTemplateId" type="anyURI">
      <annotation>
```

```xml
          <documentation>The gml:id of an previous GetObservation request response indicating observations
from a certain sensor for a certain target.
          </documentation>
        </annotation>
      </element>
      <element name="eventTime" minOccurs="0" maxOccurs="unbounded">
        <annotation>
          <documentation>Allows a client to request observations from a specific instant, multiple instances or
periods of time in the past, present and future. The supported range is listed in the selected offering capabilities.
          </documentation>
        </annotation>
        <complexType>
          <sequence>
            <element ref="ogc:temporalOps"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </extension>
 </complexContent>
 </complexType>
</element>
<!-- ================================================================

     ================================================================ -->
<element name="GetResultResponse">
  <annotation>
    <documentation>the response of a GetResult operation</documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="result">
        <annotation>
          <documentation>RS attribute points to the description of the reference system of the result. The description
will contain all information necessary to understand what is provided within the result response. The most simple
case would be a single value.</documentation>
        </annotation>
        <complexType>
          <simpleContent>
            <extension base="string">
              <attribute name="RS" type="anyURI" use="required"/>
            </extension>
          </simpleContent>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
</schema>
```

A.7 *GetFeatureOfInterest* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
    <description>This XML Schema defines the GetFeatureOfInterest request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ============================================================
     and imports
```

```
          ================================================================ -->
 <import namespace="http://www.opengis.net/ogc" schemaLocation="ogc4sos.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ================================================================

          ================================================================ -->
 <element name="GetFeatureOfInterest">
   <annotation>
     <documentation>Request to a SOS to perform the GetFeatureOfInterest operation. This operation is designed
 to request target feaure instances</documentation>
   </annotation>
   <complexType>
     <complexContent>
       <extension base="sos:RequestBaseType">
         <sequence>
           <choice>
             <element name="FeatureOfInterestId" type="anyURI" maxOccurs="unbounded">
               <annotation>
                 <documentation>Identifier of the feature of interest, for which detailed information is requested. These
 identifiers are usually listed in the Contents section of the service metadata (Capabilities) document.
 </documentation>
               </annotation>
             </element>
             <element name="location">
               <annotation>
                 <documentation>Uses modified version of filter.xsd</documentation>
               </annotation>
               <complexType>
                 <sequence>
                   <element ref="ogc:spatialOps"/>
                 </sequence>
               </complexType>
             </element>
           </choice>
           <element name="eventTime" minOccurs="0" maxOccurs="unbounded">
             <annotation>
               <documentation>Uses modified version of filter.xsd

                               a client to request targets from a specific instant, multiple instances or
 periods of time in the past, present and future.
                               is useful for dynamic sensors for which the properties of the target are time-
 dependent.
                               time paramters may be indicated so that the client may request details of the
 observation target at multiple times.
                               supported range is listed in the contents section of the service
 metadata.</documentation>
             </annotation>
             <complexType>
               <sequence>
                 <element ref="ogc:temporalOps"/>
               </sequence>
             </complexType>
           </element>
         </sequence>
       </extension>
     </complexContent>
   </complexType>
 </element>
 <!-- ====================================================

          ============================================================-->
 <!-- The response is an GML feature, the schema of the feature instance can be retrieved using the
 DescribeFeatureType request.-->
 </schema>
```

A.8 *GetFeatureOfInterestTime* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema defines the GetFeatureOfInterestTime request and response XML elements
and types.</description>
  </documentation>
 </annotation>
 <!-- ============================================================
      and imports
      ============================================================ -->
 <import namespace="http://www.opengis.net/ogc" schemaLocation="ogc4sos.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ============================================================

      ============================================================ -->
 <element name="GetFeatureOfInterestTime">
  <annotation>
   <documentation>Request to a SOS to perform the GetTargetTime operation.
          operation is designed to request the time that specified target feature instances or target locations are
available</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="FeatureOfInterestId" type="anyURI">
       <annotation>
        <documentation>Identifier of the feature of interest, for which detailed information is requested. These
identifiers are usually listed in the Contents section of the service metadata (Capabilities) document.
</documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
 </element>
 <!-- =================================================

      =========================================================-->
 <!-- The response is an GML time primitive.-->
</schema>
```

A.9 *DescribeFeatureType* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema defines the DescribeFeatureType request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ============================================================
      and imports
      ============================================================ -->
 <include schemaLocation="sosCommon.xsd"/>
```

```xml
<!-- ================================================================

     ================================================================ -->
<element name="DescribeFeatureType">
  <annotation>
    <documentation>Request to a SOS to perform the DescribeFeatureType operation. This operation is designed
to request detailed information concerning the observation's target</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="FeatureId" type="anyURI">
       <annotation>
         <documentation>Identifier of the feature for which detailed information is requested. These identifiers are
usually listed in the Contents section of the service metadata (Capabilities) document. </documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
</element>
<!-- ================================================================

     ================================================================ -->
<!--the response is a XML schema-->
</schema>
```

A.10 *DescribeObservationType* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
    <description>This XML Schema defines the DescribeObservationType request and response XML elements
and types.</description>
  </documentation>
 </annotation>
 <!-- ================================================================
      and imports
      ================================================================ -->
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ================================================================

      ================================================================ -->
 <element name="DescribeObservationType">
  <annotation>
    <documentation>Request to a SOS to perform the DescribeObservationTypeoperation. This operation is
designed to request detailed information concerning hard typed observation schemas</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="observedProperty" type="anyURI">
       <annotation>
         <documentation>The phenomenon for which the observationType (OM application schema) is
requested.</documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
```

```
    </complexContent>
  </complexType>
</element>
<!-- ================================================================

    ================================================================ -->
<!--The response will be an XML schema defining the OM applicatoin schemas-->
</schema>
```

A.11 *DescribeResultModel* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/sos/1.0" elementFormDefault="qualified" version="1.0.0"
xml:lang="en">
 <annotation>
  <documentation>
    <description>This XML Schema defines the DescribeResultModel request and response XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ================================================================
      and imports
      ================================================================ -->
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ================================================================

      ================================================================ -->
 <element name="DescribeResultModel">
  <annotation>
    <documentation>Request to a SOS to perform the DescribeResultModel operation.
          operation is designed to request detailed information concerning the format of the observation's
result</documentation>
  </annotation>
  <complexType>
   <complexContent>
    <extension base="sos:RequestBaseType">
     <sequence>
      <element name="ResultName" type="QName">
       <annotation>
         <documentation>Identifier of the type of the result, for which detailed information is
requested.</documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
 </element>
 <!-- ================================================================

      ================================================================ -->
 <!--the response is an XML schema-->
</schema>
```

A.12 *GetObservationById* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:ogc="http://www.opengis.net/ogc" xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/sos/1.0" elementFormDefault="qualified" version="1.0.0"
xml:lang="en">
 <annotation>
```

```xml
  <documentation>
    <description>This XML Schema defines the GetObservationByID request XML elements and
types.</description>
  </documentation>
 </annotation>
 <!-- ============================================================
      and imports
      ============================================================ -->
 <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
 <import namespace="http://www.opengis.net/ogc" schemaLocation="ogc4sos.xsd"/>
 <include schemaLocation="sosCommon.xsd"/>
 <!-- ============================================================


      ============================================================ -->
 <element name="GetObservationById">
  <annotation>
    <documentation>Request to a SOS to perform the GetObservation operation using an Observation
ID.</documentation>
  </annotation>
  <complexType>
    <complexContent>
     <extension base="sos:RequestBaseType">
      <sequence>
       <element name="ObservationId" type="anyURI">
        <annotation>
          <documentation>ID of the observation to obtain.  This could have been obtained by the client via a URL
in a feed, alert, or some other notification
          </documentation>
        </annotation>
       </element>
       <element name="responseFormat" type="ows:MimeType" minOccurs="0">
        <annotation>
          <documentation>ID of the output format to be used for the requested data. The supported output formats
are listed in the selected offering capabilities.
          </documentation>
        </annotation>
       </element>
       <element name="resultModel" type="QName" minOccurs="0"/>
       <element name="responseMode" type="sos:responseModeType" minOccurs="0">
        <annotation>
          <documentation>This element allows the client to request the form of the response.  The value of
resultTemplate is used to retrieve an observation template
                              will later be used in calls to GetResult.  The other options allow results to appear
inline in a resultTag (inline), external to the observation element (out-of-band)
                              as a MIME attachment (attached)</documentation>
        </annotation>
       </element>
      </sequence>
      <attribute name="srsName" type="anyURI" use="optional"/>
     </extension>
    </complexContent>
  </complexType>
 </element>
</schema>
```

A.13 *ogc4sos* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" targetNamespace="http://www.opengis.net/ogc"
elementFormDefault="qualified" version="1.0.0">
 <xs:annotation>
```

```xml
    <xs:documentation>
  ogc4sos.xs
   Utility schema which simply includes the OGC schema documents required for the OGC SOS
schemas</xs:documentation>
 </xs:annotation>
 <!-- ============================================================
     includes and imports
     ================================================================ -->
 <xs:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
 <xs:include schemaLocation="http://schemas.opengis.net/filter/1.1.0/filter.xsd"/>
 <xs:include schemaLocation="http://schemas.opengis.net/filter/1.1.0/filterCapabilities.xsd"/>
 <!-- ================================================================= -->
 <!-- TEMPORAL OPERATORS                                       -->
 <!-- ================================================================= -->
 <xs:element name="temporalOps" type="ogc:TemporalOpsType" abstract="true"/>
 <xs:complexType name="TemporalOpsType" abstract="true"/>
 <xs:element name="TM_After" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Before" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Begins" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_BegunBy" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Contains" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_During" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_EndedBy" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Ends" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Equals" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Meets" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_MetBy" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_Overalps" type="ogc:BinaryTemporalOpType" substitutionGroup="ogc:temporalOps"/>
 <xs:element name="TM_OverlappedBy" type="ogc:BinaryTemporalOpType"
substitutionGroup="ogc:temporalOps"/>
 <xs:complexType name="BinaryTemporalOpType">
   <xs:complexContent>
     <xs:extension base="ogc:TemporalOpsType">
       <xs:sequence>
         <xs:element ref="ogc:PropertyName"/>
         <xs:choice>
           <xs:element ref="ogc:PropertyName"/>
           <xs:element ref="gml:_TimeObject"/>
         </xs:choice>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="Temporal_CapabilitiesType">
   <xs:sequence>
     <xs:element name="TemporalOperands" type="ogc:TemporalOperandsType"/>
     <xs:element name="TemporalOperators" type="ogc:TemporalOperatorsType"/>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name="TemporalOperandsType">
   <xs:sequence>
     <xs:element name="TemporalOperand" type="ogc:TemporalOperandType" maxOccurs="unbounded"/>
   </xs:sequence>
 </xs:complexType>
 <xs:simpleType name="TemporalOperandType">
   <xs:restriction base="xs:QName">
     <xs:enumeration value="gml:validTime"/>
     <xs:enumeration value="gml:TimeInstant"/>
     <xs:enumeration value="gml:TimePeriod"/>
     <xs:enumeration value="gml:timePosition"/>
     <xs:enumeration value="gml:timeInterval"/>
     <xs:enumeration value="gml:duration"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:complexType name="TemporalOperatorsType">
   <xs:sequence>
```

```xml
      <xs:element name="TemporalOperator" type="ogc:TemporalOperatorType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TemporalOperatorType">
    <xs:sequence>
      <xs:element name="TemporalOperands" type="ogc:TemporalOperandsType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="ogc:TemporalOperatorNameType"/>
  </xs:complexType>
  <xs:simpleType name="TemporalOperatorNameType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="TM_After"/>
      <xs:enumeration value="TM_Before"/>
      <xs:enumeration value="TM_Begins"/>
      <xs:enumeration value="TM_BegunBy"/>
      <xs:enumeration value="TM_Contains"/>
      <xs:enumeration value="TM_During"/>
      <xs:enumeration value="TM_Equals"/>
      <xs:enumeration value="TM_Overlaps"/>
      <xs:enumeration value="TM_Meets"/>
      <xs:enumeration value="TM_OverlappedBy"/>
      <xs:enumeration value="TM_MetBy"/>
      <xs:enumeration value="TM_EndedBy"/>
      <xs:enumeration value="TM_Ends"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- ============================================ -->
  <xs:element name="Spatial_Capabilities" type="ogc:Spatial_CapabilitiesType"/>
  <xs:element name="Scalar_Capabilities" type="ogc:Scalar_CapabilitiesType"/>
  <xs:element name="Id_Capabilities" type="ogc:Id_CapabilitiesType"/>
  <xs:element name="Temporal_Capabilities" type="ogc:Temporal_CapabilitiesType"/>
</xs:schema>
```

A.15   *sosAll* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:sos="http://www.opengis.net/sos/1.0" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" version="1.0.0" xml:lang="en">
  <annotation>
    <documentation>
      <description>This XML Schema includes and imports, directly and indirectly, all the XML Schemas defined by
the OGC Sensor Observation Service (SOS).</description>
    </documentation>
  </annotation>
  <!-- ==============================================================
        and imports
        ============================================================== -->
  <include schemaLocation="sosCommon.xsd"/>
  <include schemaLocation="sosGetCapabilities.xsd"/>
  <include schemaLocation="sosDescribeFeatureType.xsd"/>
  <include schemaLocation="sosDescribeSensor.xsd"/>
  <include schemaLocation="sosDescribeObservationType.xsd"/>
  <include schemaLocation="sosDescribeResultModel.xsd"/>
  <include schemaLocation="sosGetObservation.xsd"/>
  <include schemaLocation="sosGetResult.xsd"/>
  <include schemaLocation="sosGetFeatureOfInterest.xsd"/>
  <include schemaLocation="sosGetFeatureOfInterestTime.xsd"/>
  <include schemaLocation="sosInsert.xsd"/>
  <include schemaLocation="sosRegisterSensor.xsd"/>
  <include schemaLocation="sosGetObservationById.xsd"/>
</schema>
```

A.16 *sosCommon* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:sos="http://www.opengis.net/sos/1.0" xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/sos/1.0" elementFormDefault="qualified" version="1.0.0"
xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema encodes the elements and types that are shared by muliple SOS
operations.</description>
  </documentation>
 </annotation>
 <!-- =============================================================
      and imports
      ============================================================= -->
 <!-- =============================================================
      and types
      ============================================================= -->
 <complexType name="RequestBaseType">
  <annotation>
   <documentation>XML encoded SOS operation request base, for all operations except Get Capabilities. In this
XML encoding, no "request" parameter is included, since the element name specifies the specific operation.
</documentation>
  </annotation>
  <attribute name="service" type="string" use="required" fixed="SOS">
   <annotation>
    <documentation>Service type identifier. </documentation>
   </annotation>
  </attribute>
  <attribute name="version" type="string" use="required" fixed="1.0.0">
   <annotation>
    <documentation>Specification version for SOS version and operation.</documentation>
   </annotation>
  </attribute>
 </complexType>
 <simpleType name="responseModeType">
  <restriction base="string">
   <enumeration value="inline"/>
   <enumeration value="attached"/>
   <enumeration value="out-of-band"/>
   <enumeration value="resultTemplate"/>
  </restriction>
 </simpleType>
</schema>
```

A.17 *sosContents* Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.opengis.net/sos/1.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0.0" xml:lang="en">
 <annotation>
  <documentation>
   <description>This XML Schema encodes the Contents section of the SOS GetCapabilities operation
response.</description>
```

```xml
      </documentation>
    </annotation>
    <!-- ================================================================
         and imports
         ================================================================ -->
    <import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
    <import namespace="http://www.opengis.net/swe/1.0.1"
schemaLocation="http://schemas.opengis.net/sweCommon/1.0.1/swe.xsd"/>
    <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
    <include schemaLocation="sosCommon.xsd"/>
    <!-- ================================================================
         and types
         ================================================================ -->
    <element name="Contents">
      <annotation>
        <documentation>Contents section of SOS service metadata (or Capabilites) XML document. For the SOS,
these contents are data and functions that the SOS server provides.</documentation>
      </annotation>
      <complexType>
        <sequence>
          <element name="ObservationOfferingList">
            <complexType>
              <sequence>
                <element name="ObservationOffering" type="sos:ObservationOfferingType" maxOccurs="unbounded"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
    <!-- ============================== -->
    <complexType name="ObservationOfferingBaseType" abstract="true">
      <annotation>
        <documentation>Makes boundedBy mandatory</documentation>
      </annotation>
      <complexContent>
        <restriction base="gml:AbstractFeatureType">
          <sequence>
            <group ref="gml:StandardObjectProperties"/>
            <element ref="gml:boundedBy"/>
          </sequence>
        </restriction>
      </complexContent>
    </complexType>
    <!-- ================================-->
    <complexType name="ObservationOfferingType">
      <complexContent>
        <extension base="sos:ObservationOfferingBaseType">
          <sequence>
            <element name="intendedApplication" type="token" minOccurs="0" maxOccurs="unbounded"/>
            <element name="time" type="swe:TimeGeometricPrimitivePropertyType"/>
            <element name="procedure" type="gml:ReferenceType" maxOccurs="unbounded"/>
            <element name="observedProperty" type="swe:PhenomenonPropertyType" maxOccurs="unbounded"/>
            <element name="featureOfInterest" type="gml:ReferenceType" maxOccurs="unbounded"/>
            <element name="responseFormat" type="ows:MimeType" maxOccurs="unbounded"/>
            <element name="resultModel" type="QName" minOccurs="0" maxOccurs="unbounded">
              <annotation>
                <documentation>
                            the qualified name of the observation element that will be returned from a call to
GetObservation for this offering.
                            element must be in the om:AbstractObservation substitution group and is typically
the om:Observation or a specialized extension.
                                </documentation>
              </annotation>
            </element>
```

```
    <element name="responseMode" type="sos:responseModeType" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>This element allows the client to request the form of the response.  The value of
resultTemplate is used to retrieve an observation template
                                     will later be used in calls to GetResult.  The other options allow results to appear
inline in a resultTag (inline), external to the observation element (out-of-band)
                                     as a MIME attachment (attached)</documentation>
      </annotation>
    </element>
    <!-- intendedApplication and slotID can be supported by a more generic identifier/classifier section-->
    </sequence>
   </extension>
  </complexContent>
 </complexType>
 <element name="srsName" type="gml:CodeType" substitutionGroup="ows:AbstractMetaData"/>
 <element name="supportedSensorDescription" type="QName" substitutionGroup="ows:AbstractMetaData">
  <annotation>
   <documentation>The QName of the root of a sensor desription that is supported by this service.  Examples are
"sml:_Process" and "tml:system"</documentation>
  </annotation>
 </element>
 <element name="supportedSRS" type="gml:CodeType" substitutionGroup="gml:name">
  <annotation>
   <documentation>The name by which this reference system is identified.</documentation>
  </annotation>
 </element>
</schema>
```

Annex B
(informative)

SOS Schema Examples

B.1 SOS GetCapabilities Document Example

This section contains a sample SOS GetCapabilities Document conforming to this version of this specification.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<Capabilities xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:om="http://www.opengis.net/om/1.0"
 xmlns="http://www.opengis.net/sos/1.0" xmlns:sos="http://www.opengis.net/sos/1.0"
 xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:tml="http://www.opengis.net/tml" xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
 xmlns:myorg="http://www.myorg.org/features" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosGetCapabilities.xsd "
 updateSequence="2007-06-01T11:12:39+02" version="1.0.0">
 <ows:ServiceIdentification>
  <ows:Title>MyOrg SOS</ows:Title>
  <ows:Abstract>MyOrg SOS Server</ows:Abstract>
  <ows:Keywords>
   <ows:Keyword>Weather, Chemical</ows:Keyword>
  </ows:Keywords>
  <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:SOS</ows:ServiceType>
  <ows:ServiceTypeVersion>0.0.0</ows:ServiceTypeVersion>
  <ows:Fees>NONE</ows:Fees>
  <ows:AccessConstraints>NONE</ows:AccessConstraints>
 </ows:ServiceIdentification>
 <ows:ServiceProvider>
  <ows:ProviderName>MyOrg</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://www.myorg.org"/>
  <ows:ServiceContact>
   <ows:IndividualName>John Smith</ows:IndividualName>
   <ows:PositionName>Sales Manager</ows:PositionName>
   <ows:ContactInfo>
    <ows:Phone>
     <ows:Voice>222-222-2222</ows:Voice>
     <ows:Facsimile>333-333-3333</ows:Facsimile>
    </ows:Phone>
    <ows:Address>
     <ows:DeliveryPoint>MyOrg</ows:DeliveryPoint>
     <ows:City>Chicago</ows:City>
     <ows:AdministrativeArea>IL</ows:AdministrativeArea>
     <ows:PostalCode>60606</ows:PostalCode>
     <ows:Country>USA</ows:Country>
    </ows:Address>
   </ows:ContactInfo>
  </ows:ServiceContact>
 </ows:ServiceProvider>
 <ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
   <ows:DCP>
    <ows:HTTP>
     <ows:Get xlink:href="http://www.myorg.org/sos/GetCapabilities"/>
     <ows:Post xlink:href="http://www.myorg.org/sos/GetCapabilities"/>
    </ows:HTTP>
```

```xml
      </ows:DCP>
      <ows:Parameter name="Sections">
       <ows:AllowedValues>
        <ows:Value>ServiceIdentification</ows:Value>
        <ows:Value>ServiceProvider</ows:Value>
        <ows:Value>OperationsMetadata</ows:Value>
        <ows:Value>Contents</ows:Value>
        <ows:Value>Filter_Capabilities</ows:Value>
        <ows:Value>All</ows:Value>
       </ows:AllowedValues>
      </ows:Parameter>
     </ows:Operation>
     <ows:Operation name="GetObservation">
      <ows:DCP>
       <ows:HTTP>
        <ows:Post xlink:href="http://www.myorg.org/sos/GetObservation"/>
       </ows:HTTP>
      </ows:DCP>
     </ows:Operation>
     <ows:Operation name="RegisterSensor">
      <ows:DCP>
       <ows:HTTP>
        <ows:Post xlink:href="http://www.myorg.org/sos/RegisterSensor"/>
       </ows:HTTP>
      </ows:DCP>
      <ows:Constraint name="'SupportedSensorDescription">
       <ows:AllowedValues>
        <ows:Value>sml:System</ows:Value>
        <ows:Value>tml:system</ows:Value>
       </ows:AllowedValues>
       <ows:Meaning ows:reference="urn:ogc:def:swe:SupportedSensorDescription">The service will
         only accept sensor descriptions that comply with the listed ones.</ows:Meaning>
      </ows:Constraint>
     </ows:Operation>
     <ows:Operation name="InsertObservation">
      <ows:DCP>
       <ows:HTTP>
        <ows:Post xlink:href="http://www.myorg.org/sos/InsertObservation"/>
       </ows:HTTP>
      </ows:DCP>
     </ows:Operation>
     <ows:Operation name="DescribeSensor">
      <ows:DCP>
       <ows:HTTP>
        <ows:Post xlink:href="http://www.myorg.org/sos/DescribeSensor"/>
       </ows:HTTP>
      </ows:DCP>
     </ows:Operation>
     <ows:Parameter name="service">
      <ows:AllowedValues>
        <ows:Value>SOS</ows:Value>
      </ows:AllowedValues>
     </ows:Parameter>
     <ows:Parameter name="version">
      <ows:AllowedValues>
        <ows:Value>1.0.0</ows:Value>
      </ows:AllowedValues>
     </ows:Parameter>
    </ows:OperationsMetadata>
    <Contents>
     <ObservationOfferingList>
      <ObservationOffering gml:id="offering-1">
       <gml:name>urn:MyOrg:offering:1</gml:name>
       <gml:srsName>EPSG:4326</gml:srsName>
       <gml:boundedBy>
        <gml:Envelope srsName="EPSG:4326">
         <gml:lowerCorner>39.11111 -77.17695</gml:lowerCorner>
```

```
    <gml:upperCorner>40.11111 -77.17695</gml:upperCorner>
   </gml:Envelope>
  </gml:boundedBy>
  <time>
   <gml:TimePeriod>
    <gml:beginPosition>2005-10-18T19:54:13.000Z</gml:beginPosition>
    <gml:endPosition indeterminatePosition="unknown"/>
   </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12345"/>
  <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12346"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:AirTemperature"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:AverageAirTemperature15Minute"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:WindDirection"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:Humidity"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:MaximumRelativeHumidity15Minute"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:WindSpeed"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:MaximumWindSpeed15Minute"/>
  <featureOfInterest xlink:href="urn:MyOrg:feature:station1"/>
  <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>
 </ObservationOffering>
 <ObservationOffering gml:id="offering-2">
  <gml:name>urn:MyOrg:offering:2</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
   <gml:Envelope srsName="EPSG:4326">
    <gml:lowerCorner>33.062018 -116.616526</gml:lowerCorner>
    <gml:upperCorner>33.06351 -116.61463</gml:upperCorner>
   </gml:Envelope>
  </gml:boundedBy>
  <time>
   <gml:TimePeriod>
    <gml:beginPosition>2005-10-25T14:44:10.000Z</gml:beginPosition>
    <gml:endPosition indeterminatePosition="unknown"/>
   </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12347"/>
  <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12348"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:AirTemperature"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:AverageAirTemperature15Minute"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:WindDirection"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:Humidity"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:MaximumRelativeHumidity15Minute"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:WindSpeed"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC:MaximumWindSpeed15Minute"/>
  <featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
  <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>
 </ObservationOffering>
 <ObservationOffering gml:id="offering-3">
  <gml:name>urn:MyOrg:offering:3</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
   <gml:Envelope srsName="EPSG:4326">
    <gml:lowerCorner>39.11111 -77.17695</gml:lowerCorner>
    <gml:upperCorner>40.11111 -77.17695</gml:upperCorner>
   </gml:Envelope>
  </gml:boundedBy>
  <time>
   <gml:TimePeriod>
    <gml:beginPosition>2005-10-18T19:54:13.000Z</gml:beginPosition>
    <gml:endPosition indeterminatePosition="unknown"/>
   </gml:TimePeriod>
  </time>
```

```xml
      <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
      <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12350"/>
      <observedProperty>
       <swe:CompositePhenomenon dimension="7" gml:id="AggregateChemicalPresence">
        <gml:name>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</gml:name>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGB"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGD_GF"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirVX"/>
        <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirHD"/>
       </swe:CompositePhenomenon>
      </observedProperty>
      <featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
      <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
      <resultModel>om:Observation</resultModel>
      <responseMode>inline</responseMode>
     </ObservationOffering>
     <ObservationOffering gml:id="offering-4">
      <gml:name>urn:MyOrg:offering:4</gml:name>
      <gml:srsName>EPSG:4326</gml:srsName>
      <gml:boundedBy>
       <gml:Envelope srsName="EPSG:4326">
        <gml:lowerCorner>33.062018 -116.616526</gml:lowerCorner>
        <gml:upperCorner>33.06351 -116.61463</gml:upperCorner>
       </gml:Envelope>
      </gml:boundedBy>
      <time>
       <gml:TimePeriod>
        <gml:beginPosition>2005-10-25T14:44:10.000Z</gml:beginPosition>
        <gml:endPosition indeterminatePosition="unknown"/>
       </gml:TimePeriod>
      </time>
      <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12351"/>
      <procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12352"/>
      <observedProperty xlink:href="#AggregateChemicalPresence"/>
      <featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
      <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
      <resultModel>om:Observation</resultModel>
      <responseMode>inline</responseMode>
     </ObservationOffering>
     <ObservationOffering gml:id="offering-5">
      <gml:description>Airdas 4-bands Scanner Data</gml:description>
      <gml:name>urn:MyOrg:offering:4</gml:name>
      <gml:srsName>EPSG:4326</gml:srsName>
      <gml:boundedBy>
       <gml:Null/>
      </gml:boundedBy>
      <time>
       <gml:TimePeriod>
        <gml:beginPosition>2003-11-05T19:18:58-08:00</gml:beginPosition>
        <gml:endPosition>2003-11-05T20:57:40-08:00</gml:endPosition>
       </gml:TimePeriod>
      </time>
      <procedure xlink:href="urn:ogc:object:Sensor:vast:airdas:scanner"/>
      <observedProperty>
       <swe:ConstrainedPhenomenon gml:id="BAND1">
        <gml:description>Radiance measurement in the IR band 0.61-0.68 nm</gml:description>
        <gml:name>urn:ogc:def:property:MyOrg:RADIANCE_IR_1</gml:name>
        <swe:base xlink:href="http://vast.uah.edu/dictionary/phenomena.xml#RadiationIR"/>
        <swe:singleConstraint>
         <swe:Quantity definition="urn:ogc:property:waveLength">
          <swe:uom code="um"/>
          <swe:value>0.68</swe:value>
         </swe:Quantity>
        </swe:singleConstraint>
```

```xml
        </swe:ConstrainedPhenomenon>
      </observedProperty>
      <observedProperty>
       <swe:ConstrainedPhenomenon gml:id="BAND2">
        <gml:description>Radiance measurement in the IR band 1.57-1.70 nm</gml:description>
        <gml:name>urn:ogc:def:property:MyOrg:RADIANCE_IR_2</gml:name>
        <swe:base xlink:href="http://vast.uah.edu/dictionary/phenomena.xml#RadiationIR"/>
        <swe:singleConstraint>
         <swe:Quantity definition="urn:ogc:property:waveLength">
          <swe:uom code="um"/>
          <swe:value>1.70</swe:value>
          </swe:Quantity>
        </swe:singleConstraint>
       </swe:ConstrainedPhenomenon>
      </observedProperty>
      <observedProperty>
       <swe:ConstrainedPhenomenon gml:id="BAND3">
        <gml:description>Radiance measurement in the IR band 3.60-5.50 nm</gml:description>
        <gml:name>urn:ogc:def:property:MyOrg:RADIANCE_IR_3</gml:name>
        <swe:base xlink:href="http://vast.uah.edu/dictionary/phenomena.xml#RadiationIR"/>
        <swe:singleConstraint>
         <swe:Quantity definition="urn:ogc:property:waveLength">
          <swe:uom code="um"/>
          <swe:value>5.50</swe:value>
          </swe:Quantity>
        </swe:singleConstraint>
       </swe:ConstrainedPhenomenon>
      </observedProperty>
      <observedProperty>
       <swe:ConstrainedPhenomenon gml:id="BAND4">
        <gml:description>Radiance measurement in the IR band 5.50-13.0 nm</gml:description>
        <gml:name>urn:ogc:def:property:MyOrg:RADIANCE_IR_4</gml:name>
        <swe:base xlink:href="http://vast.uah.edu/dictionary/phenomena.xml#RadiationIR"/>
        <swe:singleConstraint>
         <swe:Quantity definition="urn:ogc:property:waveLength">
          <swe:uom code="um"/>
          <swe:value>13.0</swe:value>
          </swe:Quantity>
        </swe:singleConstraint>
       </swe:ConstrainedPhenomenon>
      </observedProperty>
      <featureOfInterest xlink:href="http://vast.uah.edu/dictionary/features/WesternUS"/>
      <responseFormat>application/com-binary-base64</responseFormat>
      <resultModel>om:Observation</resultModel>
     </ObservationOffering>
    </ObservationOfferingList>
  </Contents>
</Capabilities>
```

B.2 SOS GetObservation Response Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<om:ObservationCollection xmlns:gml="http://www.opengis.net/gml"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
 xsi:schemaLocation="http://www.opengis.net/om/1.0  http://schemas.opengis.net/om/1.0.0/om.xsd"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <om:member>
  <om:Observation>
   <gml:name>urn:MyOrg:observation:1234abcde</gml:name>
   <om:samplingTime>
    <gml:TimePeriod>
     <gml:beginPosition> 2005-08-05T12:21:13Z </gml:beginPosition>
     <gml:endPosition> 2005-08-05T12:23:59Z </gml:endPosition>
    </gml:TimePeriod>
   </om:samplingTime>
```

```
    <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
    <om:observedProperty xlink:href="urn:ogc:def:property:MyOrg:AggregateChemicalPresence"/>
    <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
    <om:result>
      <swe:DataArray gml:id="ChemicalRecords">
        <swe:elementCount>
          <swe:Count>
            <swe:value>5</swe:value>
          </swe:Count>
        </swe:elementCount>
        <swe:elementType name="Components">
          <swe:SimpleDataRecord gml:id="DataDefinition">
            <swe:field name="time">
              <swe:Time definition="urn:ogc:property:time:iso8601"/>
            </swe:field>
            <swe:field name="longitude">
              <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:longitude">
                <swe:uom code="deg"/>
              </swe:Quantity>
            </swe:field>
            <swe:field name="latitude">
              <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:latitude">
                <swe:uom code="deg"/>
              </swe:Quantity>
            </swe:field>
            <swe:field name="DPM">
              <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
                <swe:uom code="ppm"/>
              </swe:Quantity>
            </swe:field>
            <swe:field name="MS">
              <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
                <swe:uom code="ppm"/>
              </swe:Quantity>
            </swe:field>
          </swe:SimpleDataRecord>
        </swe:elementType>
        <swe:encoding>
          <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
        </swe:encoding>
        <swe:values> 2005-08-05T12:21:13Z,-77.8912,38.512,20,0@@
          2005-08-05T12:22:08Z,-77.8912,38.512,30,0@@ 2005-08-05T12:22:54Z,-77.8912,38.512,0,0@@
          2005-08-05T12:23:03Z,-77.8912,38.512,10,10@@ 2005-08-05T12:23:59Z,-77.8912,38.512,10,0
        </swe:values>
      </swe:DataArray>
    </om:result>
  </om:Observation>
</om:member>
<om:member>
  <om:Observation>
    <gml:name>urn:MyOrg:observation:1234abcdf</gml:name>
    <om:samplingTime>
      <gml:TimePeriod>
        <gml:beginPosition> 2005-08-05T12:23:03Z </gml:beginPosition>
        <gml:endPosition> 2005-08-05T12:23:59Z </gml:endPosition>
      </gml:TimePeriod>
    </om:samplingTime>
    <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12350"/>
    <om:observedProperty xlink:href="urn:ogc:def:property:MyOrg:AggregateChemicalPresence"/>
    <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
    <om:result>
      <swe:DataArray gml:id="ChemValues2">
        <swe:elementCount>
          <swe:Count>
            <swe:value>2</swe:value>
          </swe:Count>
        </swe:elementCount>
```

```
   <swe:elementType name="Components" xlink:href="#DataDefinition"/>
   <swe:encoding>
     <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
   </swe:encoding>
   <swe:values> 2005-08-05T12:23:03Z,-77.8912,38.512,10,10@@
     2005-08-05T12:23:59Z,-77.8912,38.512,10,0 </swe:values>
   </swe:DataArray>
  </om:result>
 </om:Observation>
 </om:member>
</om:ObservationCollection>
```

B.3 SOS RegisterSensor Request and Response Example

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterSensor xmlns="http://www.opengis.net/sos/1.0" xmlns:tml="http://www.opengis.net/tml"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
 xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <SensorDescription>
  <tml:system>
   <tml:transducers>
    <tml:transducer>
     <tml:identification>
      <tml:description>GID3 Chemical Sensor</tml:description>
      <tml:manufacture>Smiths Detection</tml:manufacture>
      <tml:modelNumber>GID3</tml:modelNumber>
      <tml:serialNumber>333334444</tml:serialNumber>
     </tml:identification>
     <tml:transducerClass>
      <tml:transmitterReceiver>receiver</tml:transmitterReceiver>
      <tml:insituRemote>insitu</tml:insituRemote>
      <tml:spatialDependancy>attitudeIndependent</tml:spatialDependancy>
     </tml:transducerClass>
     <tml:logicalDataStructure>
      <tml:ldsDimensionality>0</tml:ldsDimensionality>
      <tml:numOfDataSetsInCf>1</tml:numOfDataSetsInCf>
      <tml:cfDataArray>
       <tml:numObjInArray>1</tml:numObjInArray>
       <tml:dataSet>
        <tml:uid>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</tml:uid>
        <tml:numObjInSet>6</tml:numObjInSet>
        <tml:dataUnit>
         <tml:uid>urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM</tml:uid>
         <tml:name>ChemicalPresenceInAirDPM</tml:name>
         <tml:dataType>number</tml:dataType>
        </tml:dataUnit>
        <tml:dataUnit>
         <tml:uid>urn:ogc:def:property:OGC:ChemicalPresenceInAirMS</tml:uid>
         <tml:name>ChemicalPresenceInAirMS</tml:name>
         <tml:dataType>number</tml:dataType>
        </tml:dataUnit>
        <tml:dataUnit>
         <tml:uid>urn:ogc:def:property:OGC:ChemicalPresenceInAirGA</tml:uid>
         <tml:name>ChemicalPresenceInAirGA</tml:name>
         <tml:dataType>number</tml:dataType>
        </tml:dataUnit>
       </tml:dataSet>
      </tml:cfDataArray>
     </tml:logicalDataStructure>
```

```xml
<tml:responseModels>
  <tml:steadyStateResponse name="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
    <tml:responseParameters>
      <tml:interCfInterpolate>lastValue</tml:interCfInterpolate>
    </tml:responseParameters>
    <tml:propValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
      <tml:valueDataType>Number</tml:valueDataType>
      <tml:values>0,9999999</tml:values>
      <tml:inputOutput>input</tml:inputOutput>
      <tml:propName>urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM</tml:propName>
      <tml:UOM>ppm</tml:UOM>
    </tml:propValues>
    <tml:dataValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
      <tml:valueDataType>Number</tml:valueDataType>
      <tml:values>0,9999999</tml:values>
      <tml:inputOutput>output</tml:inputOutput>
      <tml:dataUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM</tml:dataUidRef>
    </tml:dataValues>
  </tml:steadyStateResponse>
  <tml:steadyStateResponse name="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
    <tml:responseParameters>
      <tml:interCfInterpolate>lastValue</tml:interCfInterpolate>
    </tml:responseParameters>
    <tml:propValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
      <tml:valueDataType>Number</tml:valueDataType>
      <tml:values>0,9999999</tml:values>
      <tml:inputOutput>input</tml:inputOutput>
      <tml:propName>urn:ogc:def:property:OGC:ChemicalPresenceInAirMS</tml:propName>
      <tml:UOM>ppm</tml:UOM>
    </tml:propValues>
    <tml:dataValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
      <tml:valueDataType>Number</tml:valueDataType>
      <tml:values>0,9999999</tml:values>
      <tml:inputOutput>output</tml:inputOutput>
      <tml:dataUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirMS</tml:dataUidRef>
    </tml:dataValues>
  </tml:steadyStateResponse>
  <tml:steadyStateResponse name="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA">
    <tml:responseParameters>
      <tml:interCfInterpolate>lastValue</tml:interCfInterpolate>
    </tml:responseParameters>
    <tml:propValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
      <tml:valueDataType>Number</tml:valueDataType>
      <tml:values>0,9999999</tml:values>
      <tml:inputOutput>input</tml:inputOutput>
      <tml:propName>urn:ogc:def:property:OGC:ChemicalPresenceInAirGA</tml:propName>
      <tml:UOM>ppm</tml:UOM>
    </tml:propValues>
    <tml:dataValues>
      <tml:numValues>2</tml:numValues>
      <tml:arrayType>fcn</tml:arrayType>
      <tml:fcnInterpol>continuous</tml:fcnInterpol>
```

```
            <tml:valueDataType>Number</tml:valueDataType>
            <tml:values>0,9999999</tml:values>
            <tml:inputOutput>output</tml:inputOutput>
            <tml:dataUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirGA</tml:dataUidRef>
          </tml:dataValues>
        </tml:steadyStateResponse>
      </tml:responseModels>
      <tml:temporalModel>
        <tml:tcfTrigger>
          <tml:trigType>pvtOnChgOutput</tml:trigType>
        </tml:tcfTrigger>
        <tml:ambiguityTime>
          <tml:numValues>1</tml:numValues>
          <tml:arrayType>singleValue</tml:arrayType>
          <tml:fcnInterpol>continuous</tml:fcnInterpol>
          <tml:valueDataType>Number</tml:valueDataType>
          <tml:values>.01</tml:values>
          <tml:mult>1</tml:mult>
          <tml:offset>0</tml:offset>
          <tml:dataUidRef>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</tml:dataUidRef>
        </tml:ambiguityTime>
      </tml:temporalModel>
    </tml:transducer>
  </tml:transducers>
  <tml:clusterDescriptions>
    <tml:clusterDesc>
      <tml:description>Text</tml:description>
      <tml:idMapping>
        <tml:tapPointUidRef>ChemDataSet</tml:tapPointUidRef>
        <tml:localID>GID3</tml:localID>
      </tml:idMapping>
      <tml:clusterProperties>
        <tml:direction>fromSystem</tml:direction>
        <tml:complexity>1A</tml:complexity>
        <tml:clusterType>XML</tml:clusterType>
      </tml:clusterProperties>
      <tml:timeTag>
        <tml:accuracy>
          <tml:type>absolute</tml:type>
          <tml:errorDistribution>gaussian</tml:errorDistribution>
          <tml:factor>1sigma</tml:factor>
          <tml:accyValues>120</tml:accyValues>
        </tml:accuracy>
      </tml:timeTag>
      <tml:dataUnitEncoding name="ChemicalPresenceInAirDPM">
        <tml:dataUnitUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM</tml:dataUnitUidRef>
        <tml:dataType>Number</tml:dataType>
        <tml:dataUnitFieldSize>
          <tml:endTextDelimiter>@@</tml:endTextDelimiter>
        </tml:dataUnitFieldSize>
        <tml:encode>UTF8</tml:encode>
        <tml:numBase>10</tml:numBase>
        <tml:handleAsType>integer</tml:handleAsType>
      </tml:dataUnitEncoding>
      <tml:dataUnitEncoding name="ChemicalPresenceInAirMS">
        <tml:dataUnitUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirMS</tml:dataUnitUidRef>
        <tml:dataType>Number</tml:dataType>
        <tml:dataUnitFieldSize>
          <tml:endTextDelimiter>@@</tml:endTextDelimiter>
        </tml:dataUnitFieldSize>
        <tml:encode>UTF8</tml:encode>
        <tml:numBase>10</tml:numBase>
        <tml:handleAsType>integer</tml:handleAsType>
      </tml:dataUnitEncoding>
      <tml:dataUnitEncoding name="ChemicalPresenceInAirGA">
        <tml:dataUnitUidRef>urn:ogc:def:property:OGC:ChemicalPresenceInAirGA</tml:dataUnitUidRef>
        <tml:dataType>Number</tml:dataType>
```

```xml
        <tml:dataUnitFieldSize>
          <tml:endTextDelimiter>@@</tml:endTextDelimiter>
        </tml:dataUnitFieldSize>
        <tml:encode>UTF8</tml:encode>
        <tml:numBase>10</tml:numBase>
        <tml:handleAsType>integer</tml:handleAsType>
      </tml:dataUnitEncoding>
      <tml:numCfInCluster>1</tml:numCfInCluster>
    </tml:clusterDesc>
  </tml:clusterDescriptions>
 </tml:system>
</SensorDescription>
<ObservationTemplate>
 <om:Observation>
  <om:samplingTime>
   <gml:TimePeriod>
    <gml:beginPosition indeterminatePosition="now"/>
    <gml:endPosition indeterminatePosition="unknown"/>
   </gml:TimePeriod>
  </om:samplingTime>
  <om:procedure/>
  <om:observedProperty>
   <swe:CompositePhenomenon gml:id="composite1" dimension="3">
    <gml:name codeSpace="urn:MyOrg">urn:ogc:def:property:MyOrg:AggregateChemicalPresence</gml:name>
    <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM"/>
    <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS"/>
    <swe:component xlink:href="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA"/>
   </swe:CompositePhenomenon>
  </om:observedProperty>
  <om:featureOfInterest/>
  <om:result>
   <swe:DataArray gml:id="ChemicalRecords">
    <swe:elementCount>
     <swe:Count>
      <swe:constraint>
       <swe:AllowedValues>
        <swe:min>0</swe:min>
       </swe:AllowedValues>
      </swe:constraint>
     </swe:Count>
    </swe:elementCount>
    <swe:elementType name="Components">
     <swe:SimpleDataRecord gml:id="DataDefinition">
      <swe:field name="time">
       <swe:Time definition="urn:ogc:def:property:OGC:OriginTime"/>
      </swe:field>
      <swe:field name="latitude">
       <swe:Quantity definition="urn:ogc:def:property:OGC:Latitude:wgs84">
        <swe:uom code="deg"/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="longitude">
       <swe:Quantity definition="urn:ogc:def:property:OGC:Longitude:wgs84">
        <swe:uom code="deg"/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="altitude">
       <swe:Quantity definition="urn:ogc:def:property:OGC:Altitude:wgs84">
        <swe:uom code="m"/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="chemicalPresenceDPM">
       <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirDPM">
        <swe:uom code="ppm"/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="chemicalPresenceMS">
```

```
                 <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirMS">
                   <swe:uom code="ppm"/>
                 </swe:Quantity>
               </swe:field>
               <swe:field name="chemicalPresenceGA">
                 <swe:Quantity definition="urn:ogc:def:property:OGC:ChemicalPresenceInAirGA">
                   <swe:uom code="ppm"/>
                 </swe:Quantity>
               </swe:field>
             </swe:SimpleDataRecord>
           </swe:elementType>
           <swe:encoding>
             <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
           </swe:encoding>
           <swe:values/>
         </swe:DataArray>
       </om:result>
     </om:Observation>
   </ObservationTemplate>
 </RegisterSensor>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterSensorResponse xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0  http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <AssignedSensorId>urn:ogc:object:Sensor:MyOrg:12349</AssignedSensorId>
</RegisterSensorResponse>
```

B.4 SOS GetResult Example

Here is an example of the initial GetObservation request that creates a template for
subsequent GetResult requests. The indicator that this is a request for a result template is
in the use of the optional responseMode tag with a value of "resultTemplate".

```
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation service="SOS" version="1.0.0" xmlns="http://www.opengis.net/sos/1.0"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <offering>urn:MyOrg:offering:3</offering>
 <eventTime>
   <ogc:TM_After>
     <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
     <gml:TimeInstant>
       <gml:timePosition indeterminatePosition="now"/>
     </gml:TimeInstant>
   </ogc:TM_After>
 </eventTime>
 <procedure>urn:ogc:object:Sensor:MyOrg:12349</procedure>
 <observedProperty>urn:ogc:def:property:MyOrg:WindSpeed</observedProperty>
 <featureOfInterest>
   <ogc:BBOX>
     <ogc:PropertyName>gml:boundedBy</ogc:PropertyName>
     <gml:Envelope>
       <gml:lowerCorner>38.11 -78.6</gml:lowerCorner>
       <gml:upperCorner>38.14 -78.4</gml:upperCorner>
     </gml:Envelope>
   </ogc:BBOX>
 </featureOfInterest>
```

```
 <responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</responseFormat>
 <resultModel>om:Observation</resultModel>
 <responseMode>resultTemplate</responseMode>
</GetObservation>
```

Here is a response from an SOS with a single result template because only one sensor system was requested. The <samplingTime> element of the result set indicates that the template is valid for a period of one hour. After this time a new template will need to be requested. The identifier of the template is the gml:name attribute of the Observation within the collection. If there were more than one sensor there would be more than one template – each with its own identifier. The Observation result element and its sub-elements show how the result that is retrieved with the subsequent GetResult request should be parsed by the client.

```
<?xml version="1.0" encoding="UTF-8"?>
<om:ObservationCollection xmlns:gml="http://www.opengis.net/gml"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:swe="http://www.opengis.net/swe/1.0.1"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/om/1.0 http://schemas.opengis.net/om/1.0.0/om.xsd">
 <om:member>
  <om:Observation>
   <gml:name>urn:MyOrg:ObservationTemplate:1234abcde</gml:name>
   <om:samplingTime>
    <!-- template is valid for one hour -->
    <gml:TimePeriod>
     <gml:beginPosition>2005-08-05T12:01:32</gml:beginPosition>
     <gml:endPosition>2005-08-05T13:01:32</gml:endPosition>
    </gml:TimePeriod>
   </om:samplingTime>
   <om:procedure xlink:href="urn:ogc:object:Sensor:MyOrg:12349"/>
   <om:observedProperty xlink:href="urn:ogc:def:property:MyOrg:WindSpeed"/>
   <om:featureOfInterest xlink:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
   <om:result>
    <swe:DataArray gml:id="ChemicalRecords">
     <swe:elementCount>
      <swe:Count>
       <swe:constraint>
        <swe:AllowedValues>
         <swe:min>0</swe:min>
        </swe:AllowedValues>
       </swe:constraint>
      </swe:Count>
     </swe:elementCount>
     <swe:elementType name="Components">
      <swe:SimpleDataRecord>
       <swe:field name="time">
        <swe:Time definition="urn:ogc:property:time:iso8601"/>
       </swe:field>
       <swe:field name="longitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:longitude">
         <swe:uom code="deg"/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="latitude">
        <swe:Quantity definition="urn:ogc:property:location:EPSG:4326:latitude">
         <swe:uom code="deg"/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="windSpeed">
        <swe:Quantity definition="urn:ogc:def:property:MyOrg:WindSpeed">
         <swe:uom code="ms-1"/>
        </swe:Quantity>
```

```
        </swe:field>
      </swe:SimpleDataRecord>
    </swe:elementType>
    <swe:encoding>
      <swe:TextBlock tokenSeparator="," decimalSeparator="." blockSeparator="@@"/>
    </swe:encoding>
    <swe:values/>
   </swe:DataArray>
  </om:result>
 </om:Observation>
</om:member>
</om:ObservationCollection>
```

Here is the first GetResult request from the client. There is no time specified because all results available are requested. Note that the ObservationId matches the gml:name of the template above.

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetResult xmlns:sos="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <sos:ObservationTemplateId>urn:MyOrg:ObservationTemplate:1234abcde </sos:ObservationTemplateId>
</sos:GetResult>
```

An example of the response might be as follows. The URL in the RS attribute resolves to the contents of the template shown above.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetResultResponse xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <result RS="http://MyOrg.com/sos/templates/urn:MyOrg:ObservationTemplate:1234abcde">
  2005-08-05T12:01:32Z,33.06351,-116.61463,5.3@@ 2005-08-05T12:02:13Z,33.06351,-116.61463,5.2
 </result>
</GetResultResponse>
```

If the client made an additional request at 2005-08-05T12:20:32 and included the timestamp of the last time for which it received data the GetResult request would return all of the data since the last request. Here is an example of what that request might look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetResult xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:om="http://www.opengis.net/om/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
 service="SOS" version="1.0.0">
 <sos:ObservationTemplateId>urn:MyOrg:ObservationTemplate:1234abcde</sos:ObservationTemplateId>
 <sos:eventTime>
  <ogc:TM_After>
   <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
   <gml:TimeInstant>
    <!-- This time is the last sensor reading that was received. The client doesn't want to get duplicate records -->
    <gml:timePosition>2005-08-05T12:02:13Z</gml:timePosition>
   </gml:TimeInstant>
  </ogc:TM_After>
 </sos:eventTime>
```

```
</sos:GetResult>
```

Here is an example of what might get returned by the SOS. Note that no duplicate records are sent to the client because of the eventTime argument. If no eventTime were specified then the client would receive all records for the full hour of the template including the ones already received. The client is responsible for keeping track of state, if desired.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetResultResponse xmlns="http://www.opengis.net/sos/1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
 <result RS="http://MyOrg.com/sos/templates/urn:MyOrg:ObservationTemplate:1234abcde">
  2005-08-05T12:03:32Z,33.06351,-116.61463,5.3@@ 2005-08-05T12:04:32Z,33.06351,-116.61463,5.2@@
  2005-08-05T12:05:32Z,33.06351,-116.61463,4.7@@ 2005-08-05T12:06:32Z,33.06351,-116.61463,6.4@@
  2005-08-05T12:07:32Z,33.06351,-116.61463,4.1@@ 2005-08-05T12:08:32Z,33.06351,-116.61463,7.3@@
  2005-08-05T12:09:32Z,33.06351,-116.61463,3.8@@ 2005-08-05T12:10:32Z,33.06351,-116.61463,2.5@@
  2005-08-05T12:11:32Z,33.06351,-116.61463,4.8@@ 2005-08-05T12:12:32Z,33.06351,-116.61463,8.3@@
  2005-08-05T12:13:32Z,33.06351,-116.61463,7.2@@ 2005-08-05T12:14:32Z,33.06351,-116.61463,1.2@@
  2005-08-05T12:15:32Z,33.06351,-116.61463,3.3@@ 2005-08-05T12:16:32Z,33.06351,-116.61463,5.3@@
  2005-08-05T12:17:32Z,33.06351,-116.61463,3.6@@ 2005-08-05T12:18:32Z,33.06351,-116.61463,7.2@@
  2005-08-05T12:19:32Z,33.06351,-116.61463,7.1 </result>
</GetResultResponse>
```

Annex C
(informative)

Future Work: Issues to be Resolved

C.1 Volume and Density of Data Returned to Client

The concept of Observation Offerings allows the SOS to organize observations into groupings that are meant to be "dense" in the sense that a GetObservation request using parameters within the range identified in the offering will almost always return a non-empty collection of observations. The implementation experienced gained during the OWS-3 and OWS-4 test beds demonstrated that this was a very useful construct. Note, however, the same outcome could be obtained by factoring the offerings between SOS instances, instead of introducing this additional layer *within* a single SOS.

We also found that there is more work to be done to characterize the density of an observation offering. For example, two SOS implementations might advertise an offering that includes weather data from certain sensors over a two week period. In one instance the weather sensors are streaming real-time instantaneous measurements every three seconds. In the other instance the sensors are only reporting time-averaged data once an hour. The same request would generate widely different record counts on the two service instances. The goal for SOS is to allow clients to discover and request observations from SOS instances with no a-priori knowledge so there is a missing piece that would allow a client to determine the density of an offering. Also, in the case of the SOS that reports real-time weather data every three seconds, the client might want to receive data only every hour. The client might accept hourly data that is just an arbitrary sampling of the raw data or might prefer time-averaged data or statistical summaries of the data that would allow it to see outliers such as a high wind speed value that resulted from a tornado that came near the weather station for 5 minutes sometime during the hour.

It may or may not be appropriate for an SOS to provide time-averaging or statistical computations on raw sensor data. Perhaps a new or existing service could handle that type of functionality. The density of requests and offerings must be addressed, however, so that large data volumes are not transmitted unnecessarily due to a lack of information about service offerings.

Example 1 – Possible approach to observation density

Here is one approach that could be used to specify density in a GetObservation request. In this example, the client is making a request for data at regular time intervals of one minute starting at 2006-12-07T15:45:00.   Since the time interval is one minute and the

request is for 100 time units the last observation received should be no later than 2006-12-07T17:25:00, which is 100 minutes (1 hour and 40 minutes) later. The client may receive no more than 100 records in response to this request. If the available data is sparser than one observation per minute then the client will get less than 100 observations.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservation xmlns:sos="http://www.opengis.net/sos/1.0"
 xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengis.net/ows/1.1"
 xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:om="http://www.opengis.net/om/1.0"
 xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd
http://www.opengis.net/swe/1.0.1 http://schemas.opengis.net/sweCommon/1.0.1/swe.xsd"
 service="SOS" version="1.0.0">
 <sos:offering>urn:MyOrg:offering:3</sos:offering>
 <sos:eventTime>
  <ogc:TM_During>
   <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
   <swe:TimeInstantGrid>
    <swe:extent>
     <swe:TimeGridEnvelope>
      <swe:low>0</swe:low>
      <swe:high>100</swe:high>
     </swe:TimeGridEnvelope>
    </swe:extent>
    <swe:originPos>2006-12-07T15:45:00</swe:originPos>
    <!-- one minute -->
    <swe:offsetDuration>PT1M</swe:offsetDuration>
   </swe:TimeInstantGrid>
  </ogc:TM_During>
 </sos:eventTime>
 <sos:observedProperty>urn:ogc:def:property:MyOrg:AggregateChemicalPresence</sos:observedProperty>
 <sos:responseFormat>text/xml; subtype=&quot;om/1.0.0&quot;</sos:responseFormat>
 <sos:resultModel>om:Observation</sos:resultModel>
 <sos:responseMode>inline</sos:responseMode>
</sos:GetObservation>
```

C.2 Sensor Identifiers

Sensor identifiers are important within SOS and the SWE initiative because some users want to access data in a sensor-centric fashion. These users are already familiar with particular sensors and want to discover what services can return data from those sensors. Because of the wide variety of scenarios in which SWE technologies will be used and because of the diversity of mechanisms for assigning identifiers to sensors, it is not feasible to define a new globally unique system for assigning identifiers to sensors. The assignment of identifiers to sensors is inherently decentralized just as is the definition of phenomena discussed in a previous section.

Every sensor identifier is associated with some authority that controls the assignment of identifiers. Each of these authorities has its own mechanism for generating and assigning identifiers to sensors. In some cases the identifier may come from the serial number of the sensor that was assigned by the manufacturer. In other cases the identifier may be an arbitrary unique value. In some cases the namespace used for identifiers will be flat and in others it will follow some kind of hierarchy.

One approach for representing sensor identifiers that was considered during the OWS-3 test bed was to define an extension to OGC document 05-010 - URNs of Definitions in OGC Namespace. This extension would add a category label, resource group, resource type, and resource subtype to the existing OGC URN format. The general form would be

```
urn:ogc:category.label:resource.group:resource.type-
resource.subtype:authority:version:code
```

We could add an additional category.label "object" and resource.group "feature" (or maybe collapse these to just "feature") and use the resource.type for the featureType, in this case with the value "Sensor", and then we would need to define the parameters. Following the pattern of the "def" profile, the URN values might look something like this:

urn:ogc:object:Sensor:NASA:version:code

urn:ogc:object:Sensor-DewPointTemperature:3eTI:234567ABCD

urn:ogc:object:Sensor-Elevation:3eTI:987etgh345mnbvx

urn:ogc:object:Sensor:IRIS:WS01:12345

urn:ogc:object:Sensor:IRIS:MOBIL01:12345

urn:ogc:object:Sensor:IRIS:IRIS01_CAMERA:12345

The sensor-subtype would be optional, but if present would be taken from the OGC canonical phenomenon dictionary. The tail of the identifier, after the authority (e.g., IRIS, NASA, 3eTI), is defined by the provider and is usually opaque to the user. There could be additional colon-delimited fields if the authority uses a hierarchical namespace. A list of valid authorities would be maintained in an OGC dictionary.

Another issue regarding sensor identifiers is whether the identifier refers to the sensor or the output of the sensor. Some sensor systems have more than one output. For example the R.M. Young model 05103-5 Wind Monitor reports both wind speed and wind direction. This sensor has two analog outputs: one for each measured phenomena. An identifier could be assigned to the sensor as a whole, to each of the outputs, or to both the sensor and the outputs. In a subsequent version of this specification we should define whether a reference to the identifier for this sensor in a capabilities document or a GetObservation request implies a reference to both sensor outputs.

C.3 Aggregate and Summary Phenomena

Aggregate and summary phenomena are commonly used in sensor applications. An example of a sensor that measures an aggregate phenomenon is one that measures radiance by measuring several discrete wavelengths that are used to construct a radiance band. Another example is an earthquake location sensor that reports epicentre, depth, and origin time. The observations that capture these aggregate phenomena could be broken into their constituent parts. However, the application domains in which they are used

generally require that they be reported as if they were base phenomena. It might not make sense to report each wavelength value separately or to report only two of the earthquake location components, for example.

Summary phenomena are commonly used in meteorology. Weather data often includes things like 15 minute averaged temperature and one hour averaged temperature in addition to instantaneous temperature. In most metrological applications the instantaneous values are not useful because the models require time-averaged values. Data loggers on weather stations frequently calculate the summary data points internally and report them as if they were raw sensor data. In this case the raw values of temperature, wind speed, humidity, etc. are probably not even recoverable since they are only stored temporarily in the data logger for computing the summary data. Theoretically these data could be split into their constituent values. However, just as for the aggregate phenomena discussed above, the application domain requires that the summary data be considered as base phenomena.

We must be careful with how we characterize aggregate and summary observations in the context of the SWE initiative. In general we can characterize these observations as capturing aggregate or summary phenomena directly or we can characterize them as observations that capture base phenomena and which can be transformed through a SensorML process chain or TML transducer characterizations to obtain aggregate or summary observations. In the former case we would explicitly define aggregate or summary phenomena in a dictionary and advertise sensors that generated observations capturing those phenomena. The Observation Offering would include fewer sensor systems and more phenomena (i.e., observedProperty tags). In the latter case we would advertise different sensor systems (i.e., procedure tags) that are actually different outputs from the same sensor which were derived from raw sensor data using a SensorML or TML description. The Observation Offering would include more sensor systems and fewer phenomena.

Flexibility in how observations can be represented is a two-edged sword. Flexibility allows different sensor communities and service providers to advertise capabilities in a way that makes sense for them. However, it also means that essentially the same offering can be encoded in multiple ways. This can lead to interoperability problems because it would be difficult for a client to determine which service offerings are equivalent.

In a future version of the specification we should provide guidance for how to handle aggregate and summary observations.

Annex D
(informative)
UML Model

D.1 Introduction

This annex provides a UML model of the SOS interface, using the OGC/ISO profile of UML summarized in Subclause 5.2.

Figure D.1 is a UML diagram summarizing the SOS interface. This class diagram shows that the SensorObservationServer class inherits the getCapabilitites operation from the abstract OGCWebService class, which is common to all OGC Web Services. The SensorObservationService class adds the getObservation and describeSensor operations. Further specialization of SensorObservationServer is defined by the EnhancedSensorObservationServer and TransactionSensorObservationServer interfaces.
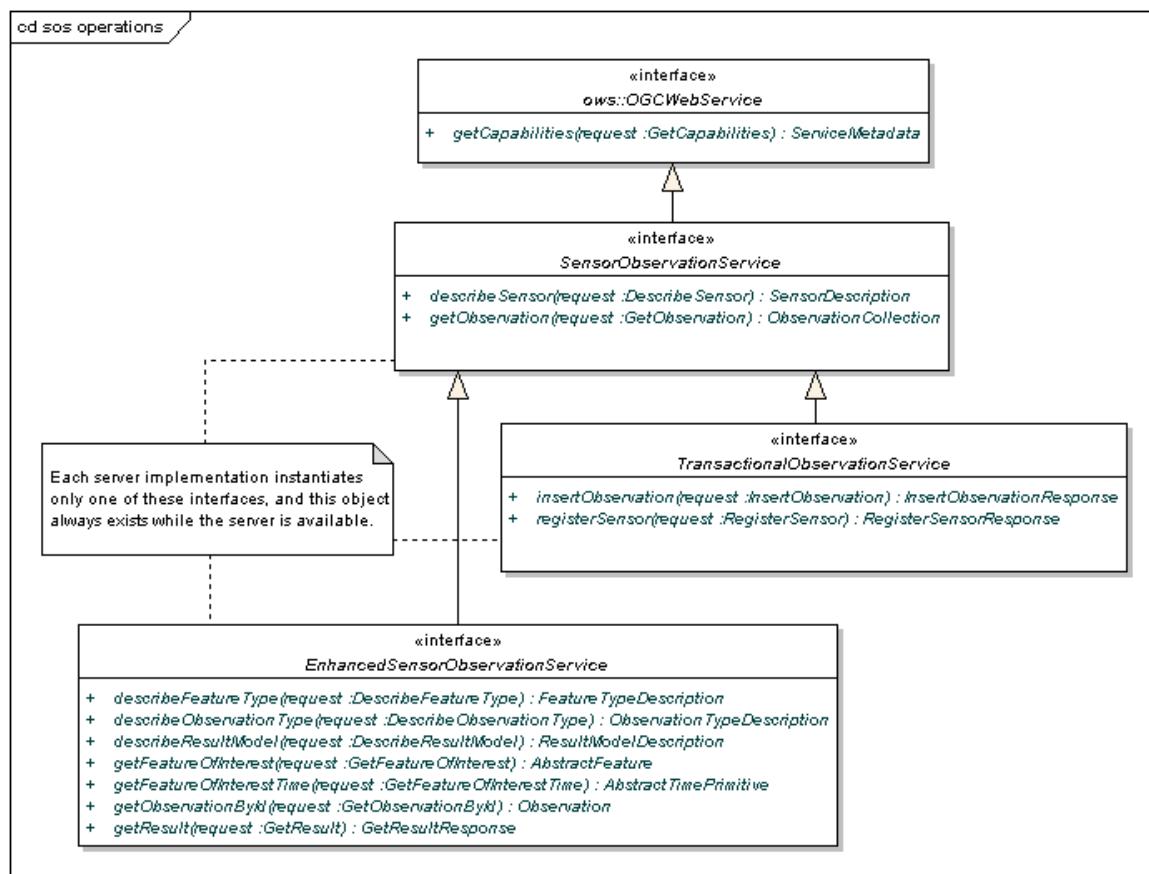

Figure D. 1 SOS interface in UML notation

All operations use a request and response data type, each of which can also be defined by one or more additional UML classes.

Bibliography

[1] ISO 31 (all parts), Quantities and units.

[2] IEC 60027 (all parts), Letter symbols to be used in electrical technology.

[3] ISO 1000, SI units and recommendations for the use of their multiples and of certain other units.

[4] Template for OWS Implementation Specifications, OGC document 05-009r1.

[5] Observations and Measurements, OGC document 07-022r1.

[6] Sensor Model Language (SensorML) for In-situ and Remote Sensors, OGC document 07-000.

[7] Transducer Markup Language (TML), OGC document 06-010r6.

[8] Sensor Planning Service (SPS), OGC document 07-014r3.

[9] Web Notification Service, OGC document 06-095r1.

[10]   Sensor Alert Service, OGC document 06-028r4.

[11]   Sensor Web Enablement Architecture, OGC document 06-050r2.

[12]   OpenGIS® Web Services Common Specification, OGC document 06-121r3