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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/

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

Contents

1 Scope....1
2 Conformance....1
3 Normative references....1
4 Terms and definitions....2
5 Conventions....3
5.1 Symbols (and abbreviated terms)....3
5.2 UML Notation....4
5.3 XML schema notation....4
5.3.1 Element....4
5.3.2 Optional Element....5
5.3.3 Recurring Element....5
5.3.4 Sequence Connector....5
5.3.5 Choice Connector....6
5.3.6 Definition with Complex Type....6
5.3.7 Complex Type....7
6 SOS Overview....8
6.1 General Approach....8
6.2 The Observation model....8
6.3 Observation Offerings....9
6.4 Identifying and referencing Phenomena and Units of Measure....10
6.5 Describing and defining Phenomena and Units of Measure....11
6.6 Filtering and Filter Expressions....12
7 SOS Concept of Operations....12
7.1 Overview....12
7.2 Concept of Operations for Sensor Data Consumer....13
7.2.1 Introduction....13
7.2.2 Service Discovery....16
7.2.3 Observation Discovery....16
7.2.4 Get Sensor Metadata....16
7.2.5 Get Sensor Observations....16
7.3 Concept of Operations for Sensor Data Publisher....17
7.3.1 Introduction....17
7.3.2 Service Discovery....20
7.3.3 Sensor Registration....20
7.3.4 Publishing of Observations....20
8 Core Operations Profile....20
8.1 Overview....20
8.2 GetCapabilities (mandatory)....21

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

8.2.1 Introduction....21
8.2.2 Request....21
8.2.3 Response....21
8.2.4 Exceptions....24
8.2.5 Examples....25
8.3 DescribeSensor (mandatory)....25
8.3.1 Introduction....25
8.3.2 Request....25
8.3.3 Response....26
8.3.4 Exceptions....26
8.3.5 Examples....26
8.4 GetObservation (mandatory)....27
8.4.1 Introduction....27
8.4.2 Request....27
8.4.3 Response....30
8.4.4 Exceptions....30
8.4.5 Examples....30

9 Transaction Operations Profile (optional)....33
9.1 RegisterSensor (optional)....33
9.1.1 Introduction....33
9.1.2 Request....33
9.1.3 Response....35
9.1.4 Exceptions....35
9.1.5 Examples....35
9.2 InsertObservation (optional)....35
9.2.1 Introduction....35
9.2.2 Request....35
9.2.3 Response....36
9.2.4 Exceptions....36
9.2.5 Examples....36

10 Enhanced Operations Profile (optional)....38
10.1 GetObservationById (optional)....38
10.1.1 Introduction....38
10.1.2 Request....38
10.1.3 Response....39
10.1.4 Exceptions....39
10.1.5 Examples....40
10.2 GetResult (optional)....41
10.2.1 Introduction....41
10.2.2 Request....42
10.2.3 Response....42
10.2.4 Exceptions....43
10.2.5 Examples....43
10.3 GetFeatureOfInterest (optional)....43
10.3.1 Introduction....43
10.3.2 Request....43
10.3.3 Response....44

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

10.3.4 Exceptions **44**
10.3.5 Examples **44**
10.4 GetFeatureOfInterestTime (optional) **45**
10.4.1 Introduction **45**
10.4.2 Request **45**
10.4.3 Response **46**
10.4.4 Exceptions **46**
10.4.5 Examples **46**
10.5 DescribeFeatureType (optional) **47**
10.5.1 Introduction **47**
10.5.2 Request **47**
10.5.3 Response **47**
10.5.4 Exceptions **47**
10.5.5 Examples **48**
10.6 DescribeObservationType (optional) **49**
10.6.1 Intoduction **49**
10.6.2 Request **49**
10.6.3 Response **49**
10.6.4 Exceptions **49**
10.6.5 Examples **50**
10.7 DescribeResultModel (optional) **50**
10.7.1 Introduction **50**
10.7.2 Request **51**
10.7.3 Response **51**
10.7.4 Exceptions **51**
10.7.5 Examples **52**
C.1 Volume and Density of Data Returned to Client **85**
C.2 Sensor Identifiers **86**
C.3 Aggregate and Summary Phenomena **87**
D.1 Introduction **89**

Figures Page

Figure 1-1 General Case for In-Situ Sensors ..... **xiii**
Figure 5-1. Simple XML Spy element **5**
Figure 5-2. Optional XML Spy element **5**
Figure 5-3. Recurring XML Spy element **5**
Figure 5-4. Sequence Connector in XML Spy **6**
Figure 5-5. Choice Connector in XML Spy **6**
Figure 5-6. Definition with Complex Element in XML Spy **7**
Figure 5-7. Complex Type in XML Spy **7**
Figure 6-1. O&M Model Exract **9**

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Figure 7-1 Sensor Data Consumer in Operational Context 13
Figure 7-2 Sensor Data Consumer Flow Chart 14
Figure 7-3 Sensor Data Consumer Sequence Diagram 15
Figure 7-4 Operational Context for Sensor Data Producers 18
Figure 7-5 Sensor Data Producer Flow Chart 18
Figure 7-6 Sensor Data Producer Sequence Diagram 19
Figure 8-1. GetCapabilities Response 22
Figure 8-2 Observation Offering 23
Figure 8-3 DescribeSensor Request 25
Figure 8-4 GetObservation Request 28
Figure 9-1 Parameters of a RegisterSensor Request 34
Figure 9-2 Parameters of an InsertObservation Request 36
Figure 10-1 GetObservationById Request 38
Figure 10-2 Parameters of GetResult Request 42
Figure 10-3 Parameters for a GetFeatureOfInterest Request 43
Figure 10-4 Parameters for a GetFeatureOfInterestTime Request 45
Figure 10-5 Parameters for DescribeFeatureType Request 47
Figure 10-6 Parameters for a DescribeObservationType Request 49
Figure 10-7 Parameters for the DescribeResultModel Request 51

Tables Page

Table 1 — Additional Section name values and meanings 21
Table 2 Observation Offering Constraints 24
Table 3 Attributes of DescribeSensor Request 26
Table 4 Parameters of GetObservation Request 29
Table 5 Parameters of RegisterSensor Request 34
Table 6 Parameters of insertObservation Request 36
Table 7 Parameters of GetObservation Request 39
Table 8. Parameters of GetResult Request 42
Table 9. Parameters of GetFeatureOfInterest Request 44
Table 10. Parameters of GetFeatureOfInterestTime Request 46
Table 11. Parameters of DescribeFeatureType Request 47
Table 12 Parameters of DescribeObservationType Request 49
Table 13 Parameters of DescribeResultModel Request 51

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

				(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)]

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

- 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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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 (S*n*) of various types are grouped into several constellations (C*n*) that are then accessed through some service (e.g., SOS).

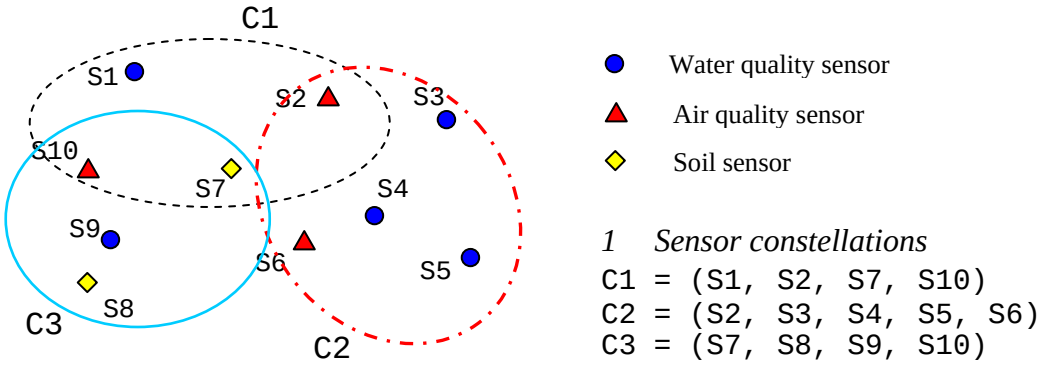

Figure 1-1 General Case for In-Situ Sensors

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

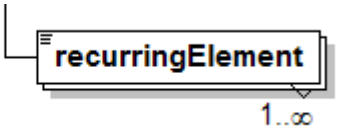

Figure 5-3. Recurring XML Spy element

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

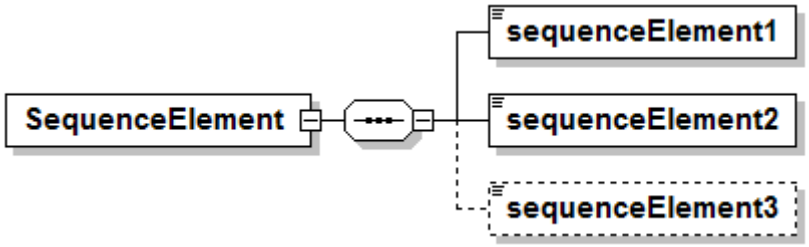

Figure 5-4. Sequence Connector in XML Spy

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.

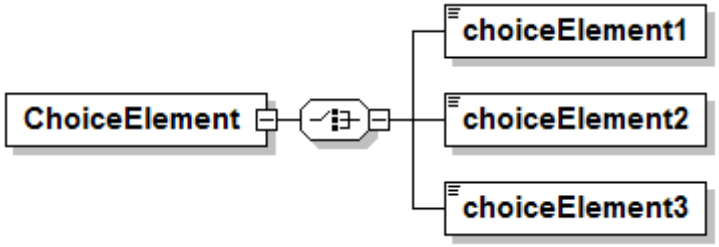

Figure 5-5. Choice Connector in XML Spy

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”).

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

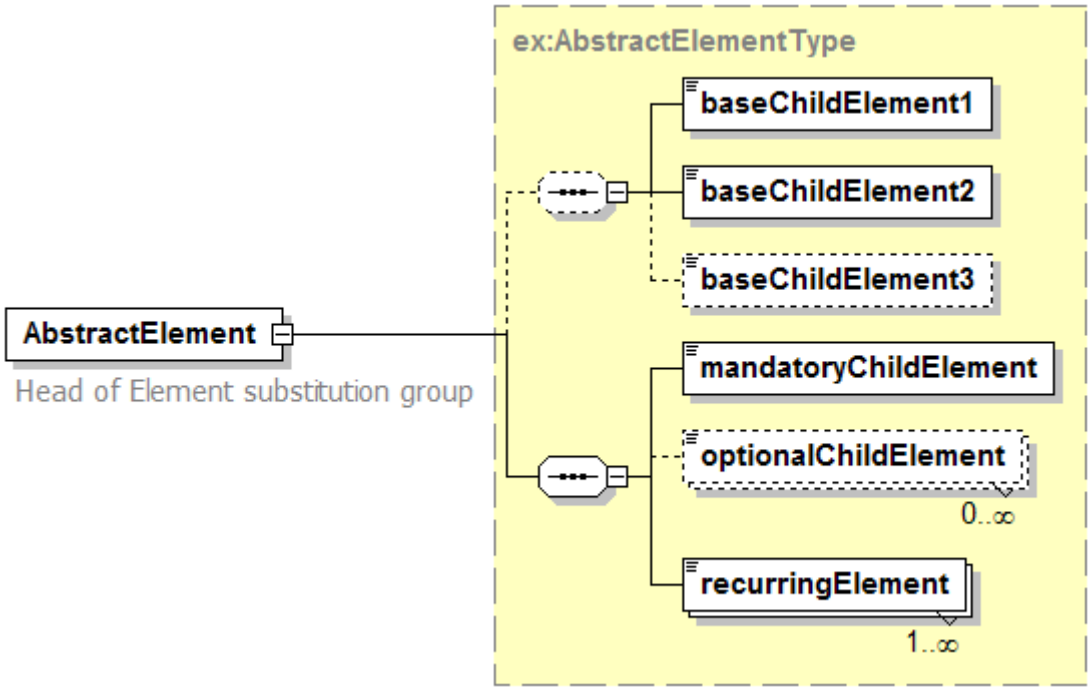

Figure 5-6. Definition with Complex Element in XML Spy

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.

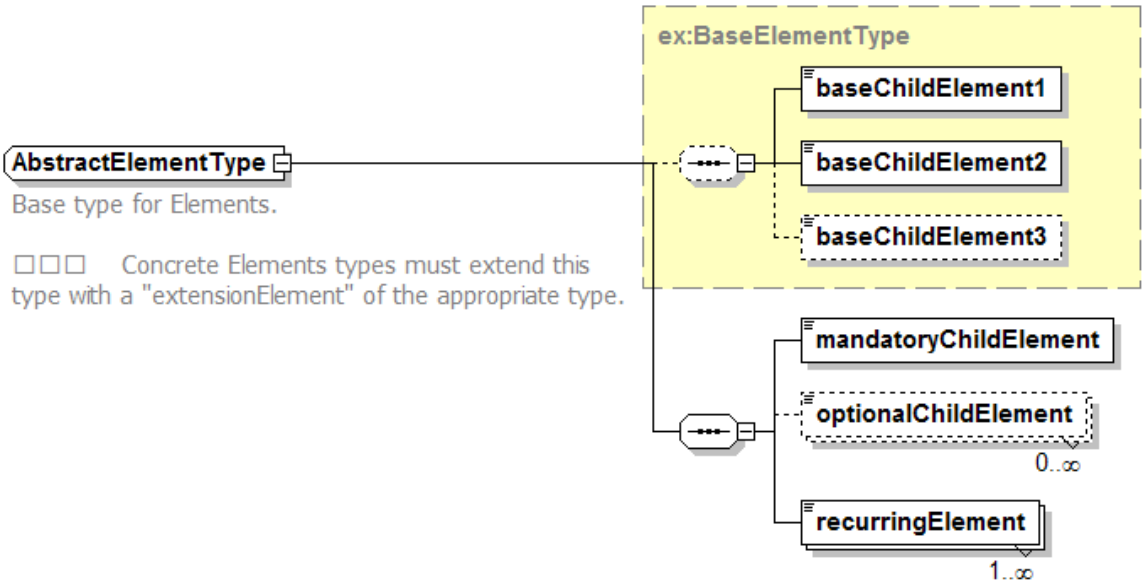

Figure 5-7. Complex Type in XML Spy

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

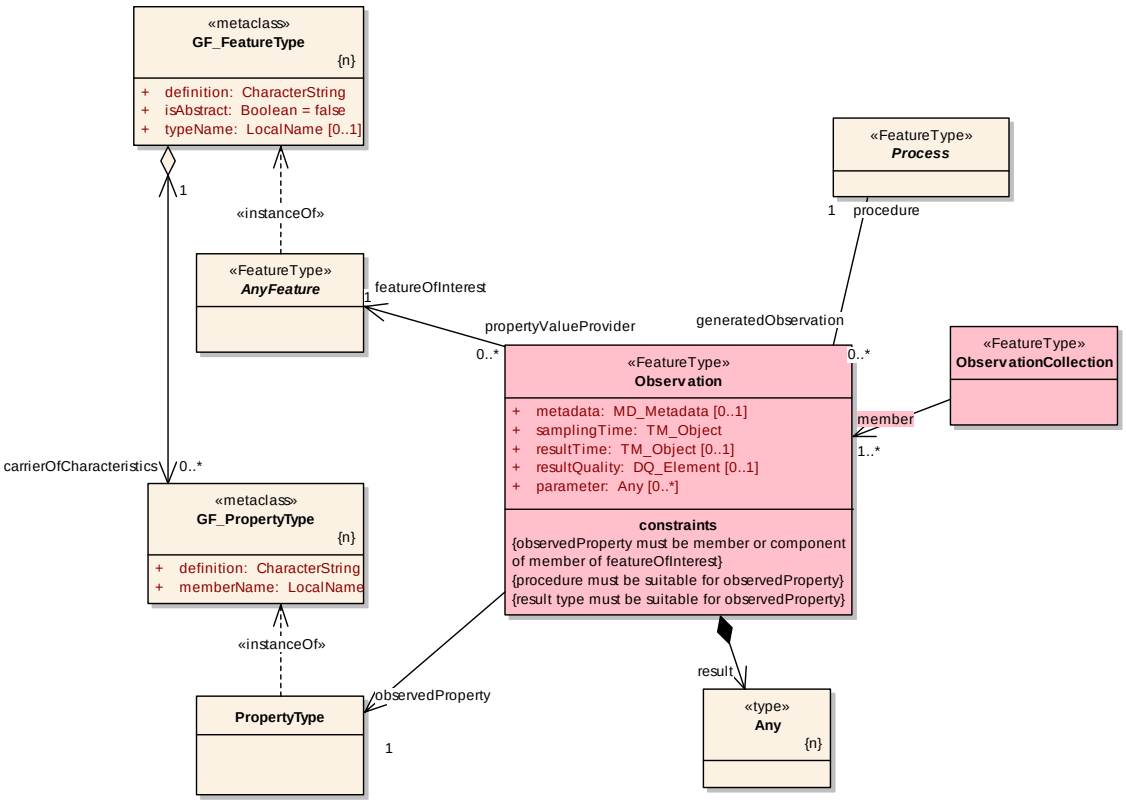

Figure 6-1. O&M Model Exract

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

- 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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

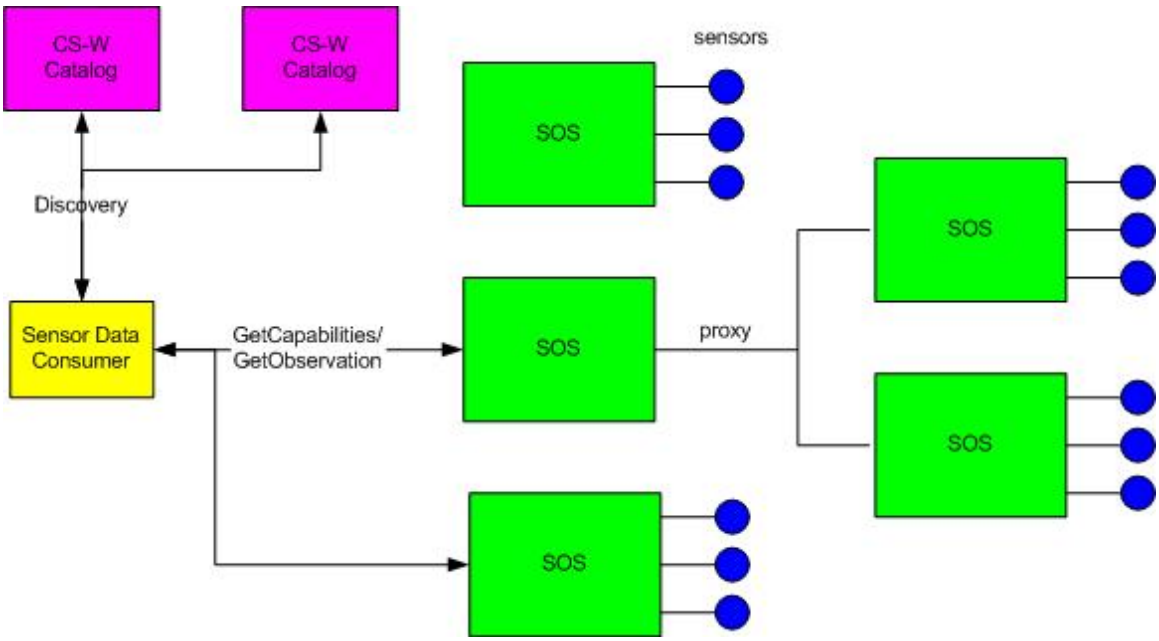

Figure 7-1 Sensor Data Consumer in Operational Context

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Figure 10 is a flow chart that shows the mandatory and optional steps that can be followed by the sensor data consumer.

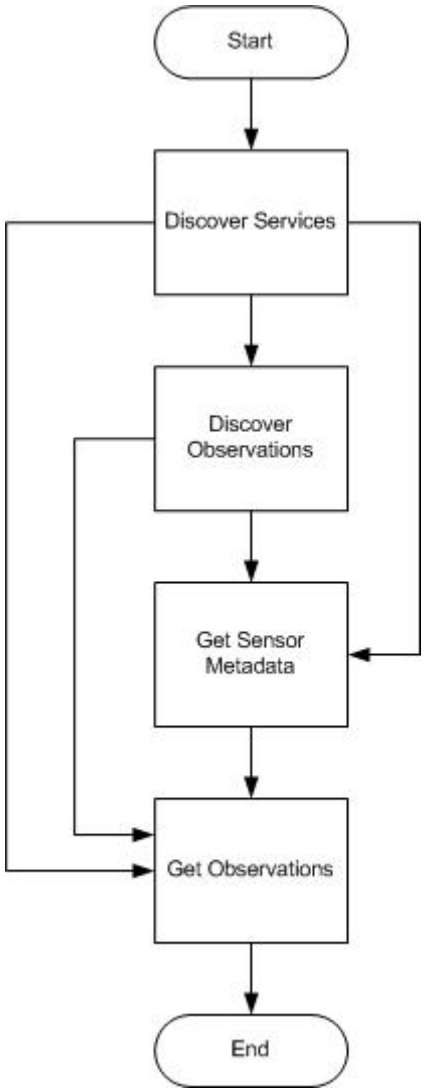

Figure 7-2 Sensor Data Consumer Flow Chart

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

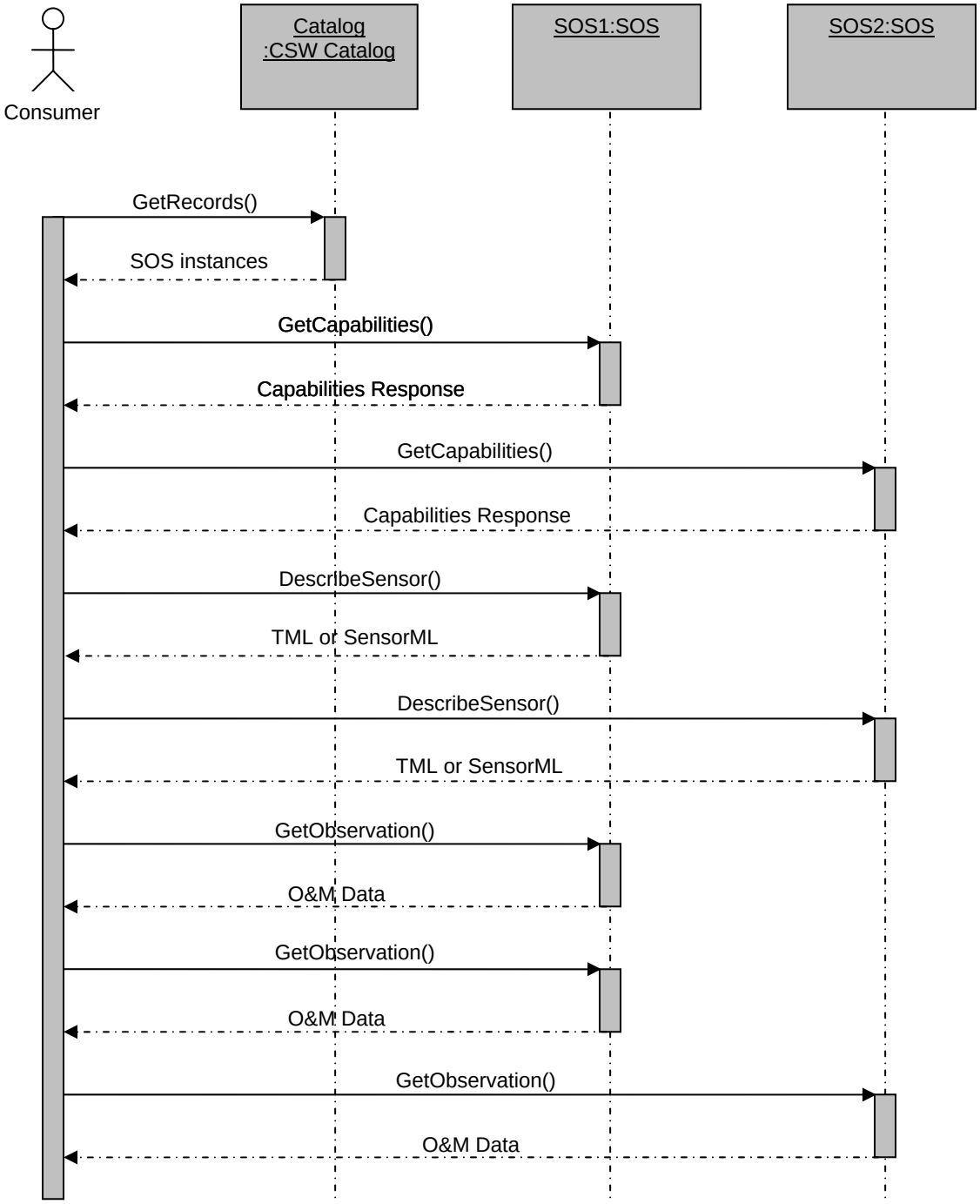

Figure 7-3 Sensor Data Consumer Sequence Diagram

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

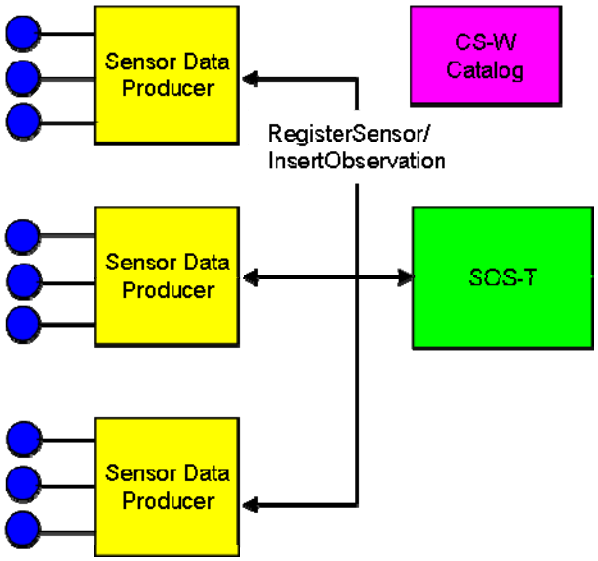

Figure 7-4 Operational Context for Sensor Data Producers

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.

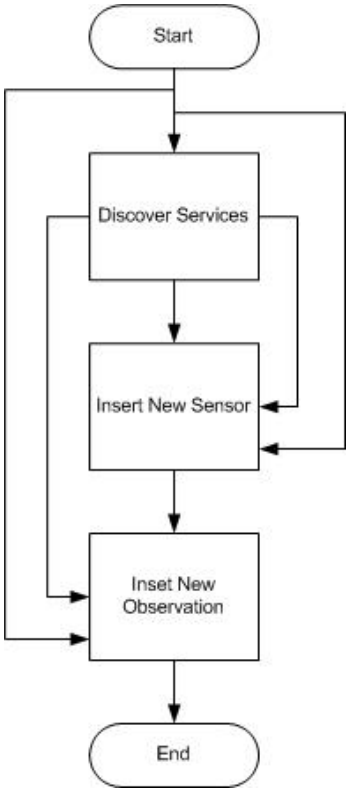

Figure 7-5 Sensor Data Producer Flow Chart

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

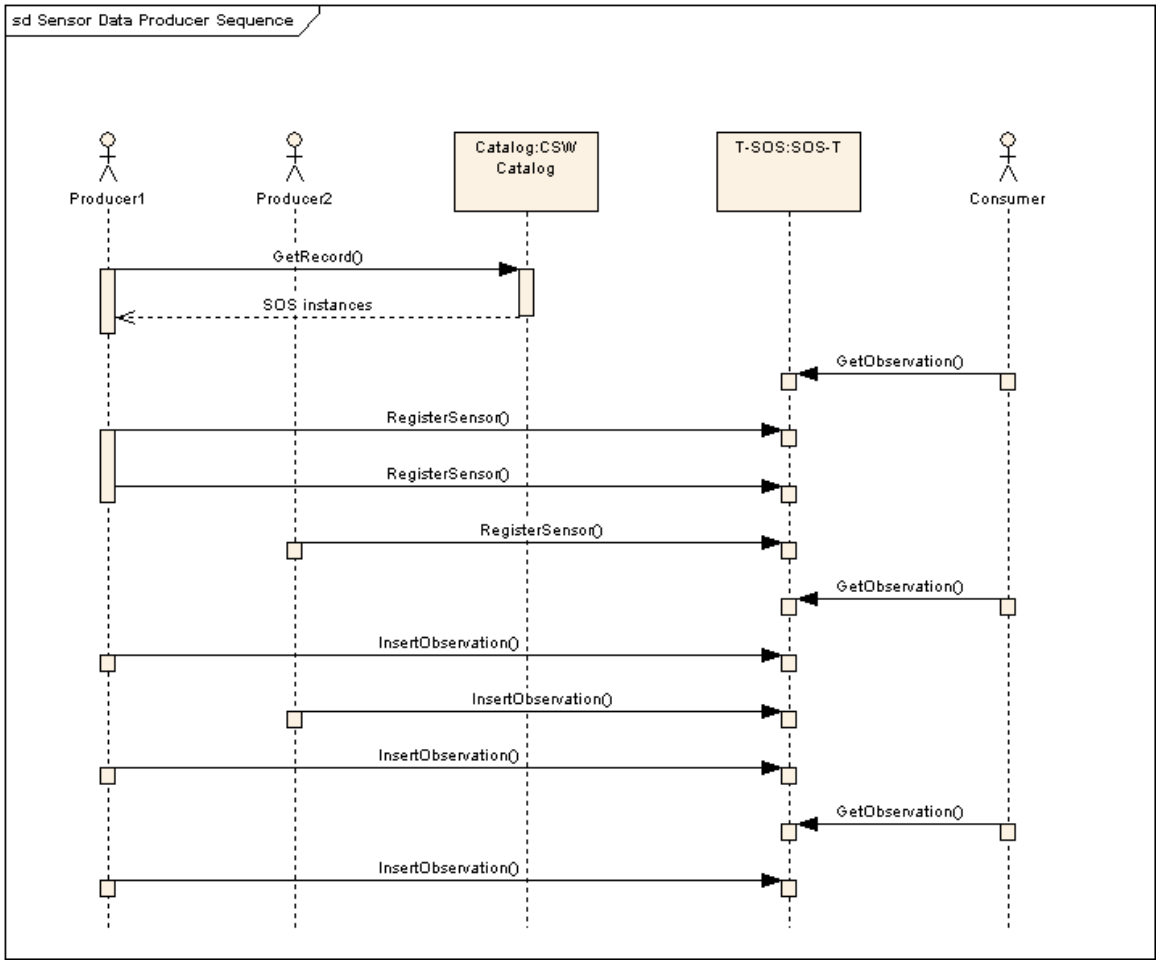

Figure 7-6 Sensor Data Producer Sequence Diagram

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

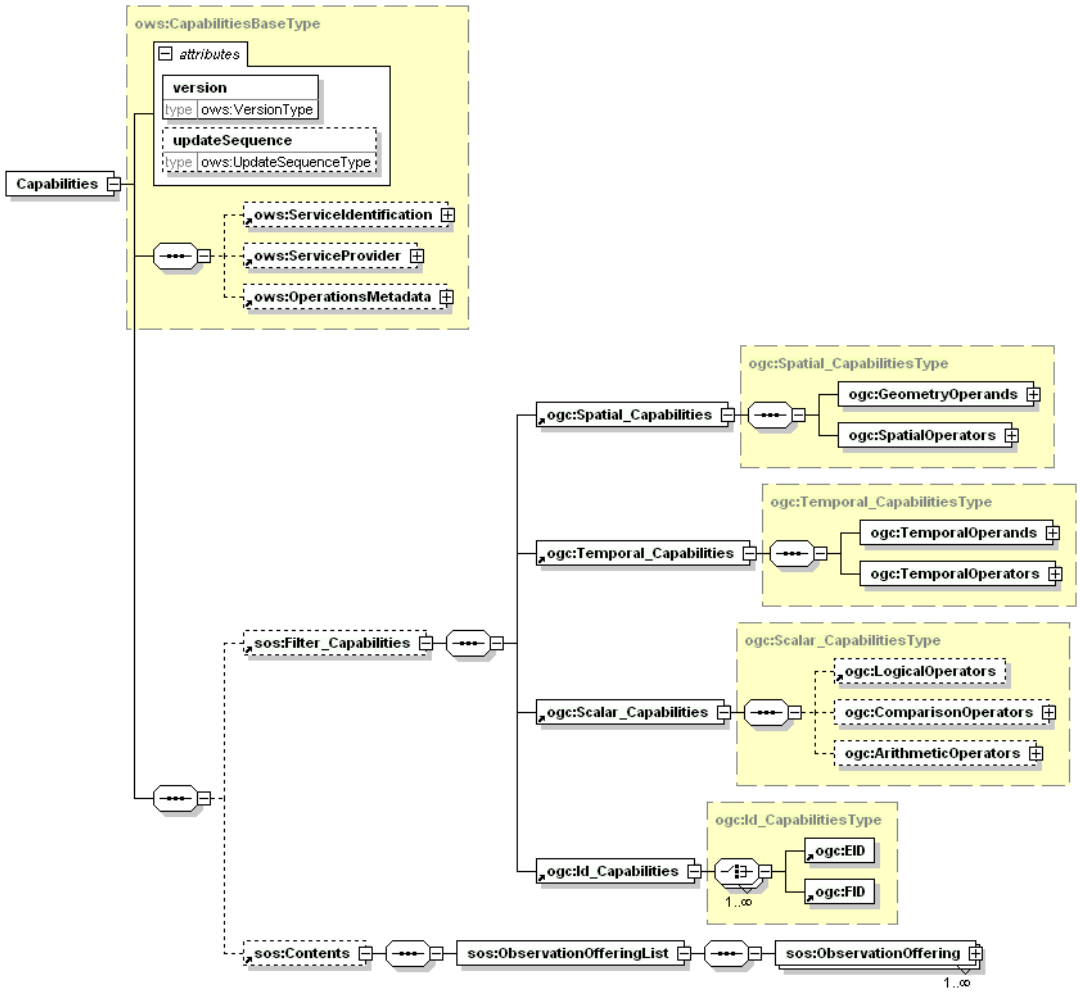

Figure 8-1. GetCapabilities Response

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

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

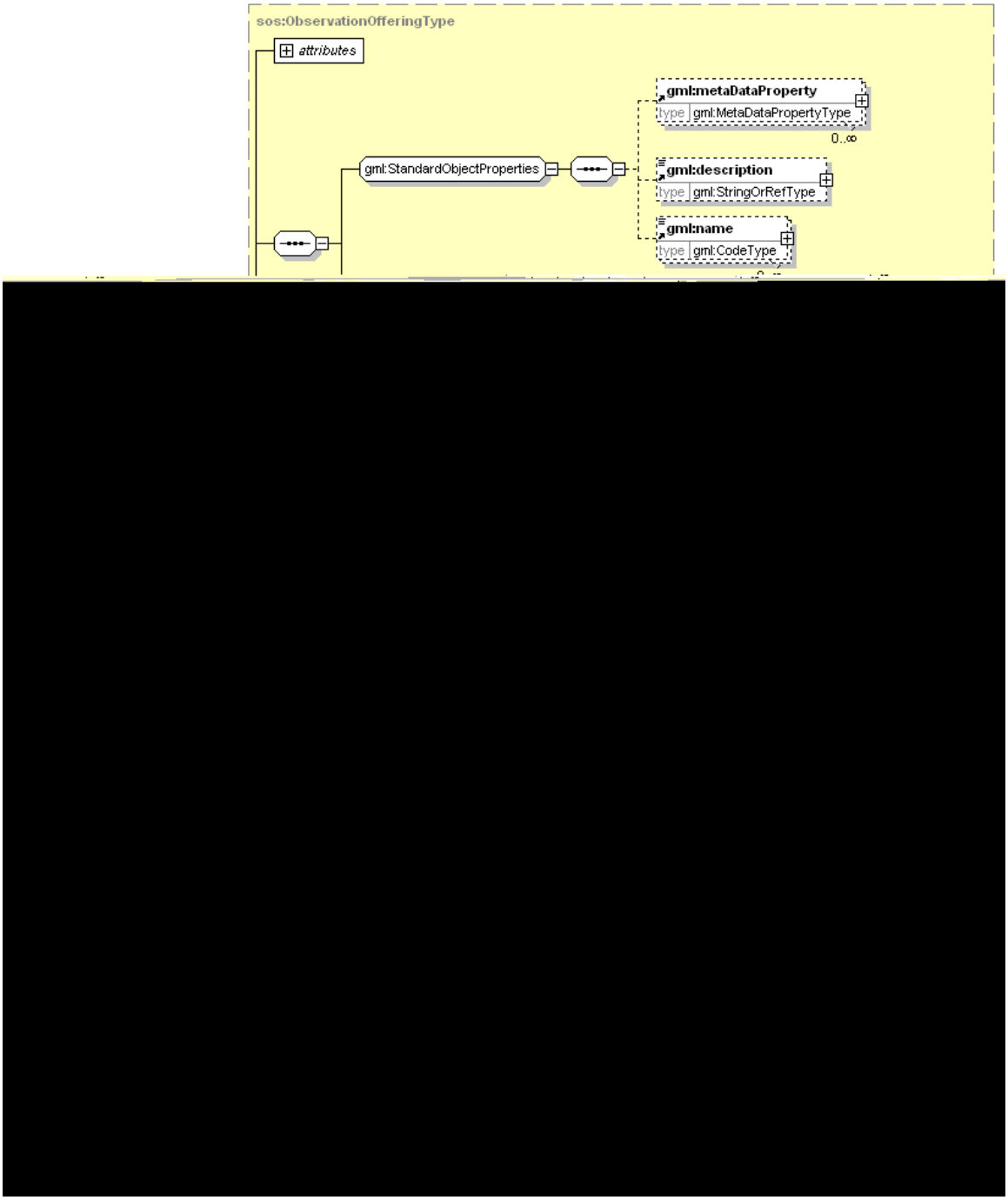

Figure 8-2 Observation Offering

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      </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 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>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
    <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>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      <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>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
<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>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
          <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>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
        <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">
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
            <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>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
 <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>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
          </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>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
</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>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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 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=&amp;quot;om/1.0.0&amp;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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

 Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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.

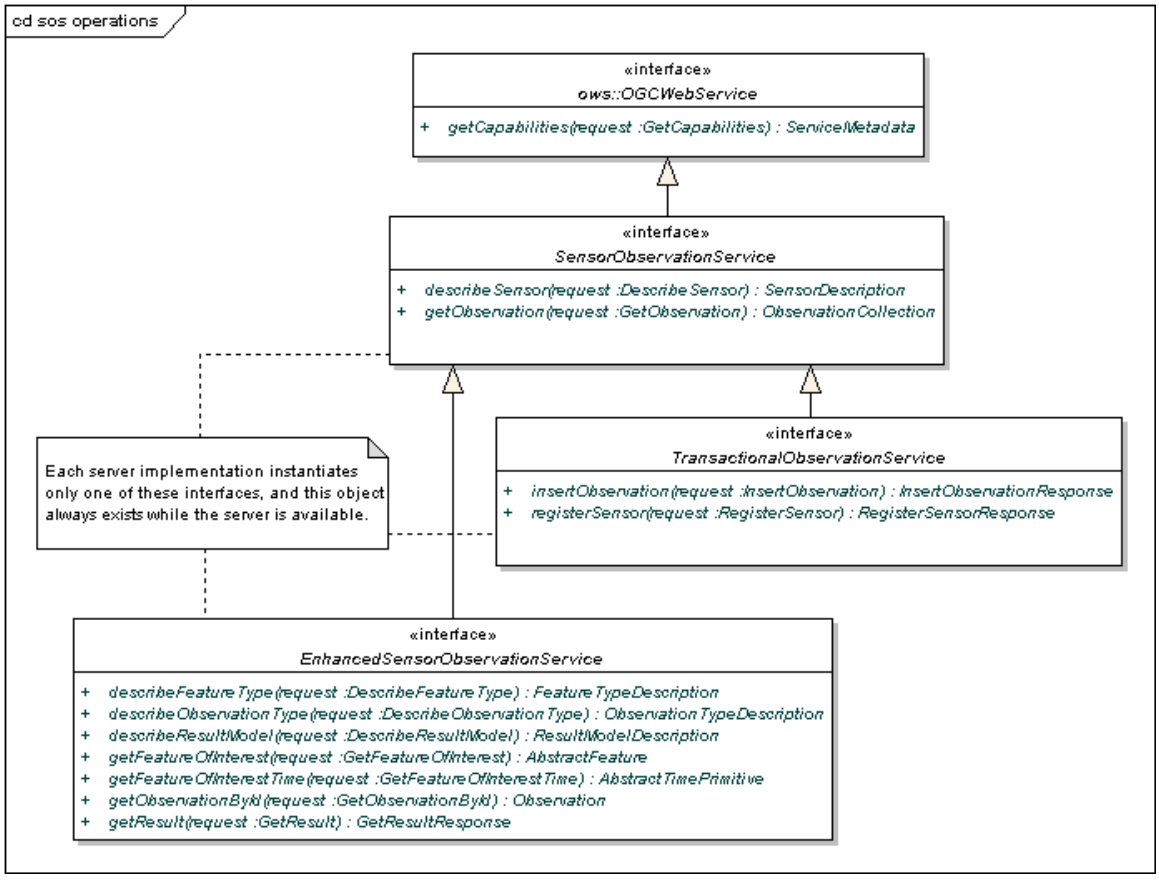

Figure D. 1 SOS interface in UML notation

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

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

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.