Open Geospatial Consortium Inc.

Date: 2008-10-10

Reference number of this document: OGC 08-133

Version: 0.3.0

Category: OpenGIS® Discussion Paper

Editors: Johannes Echterhoff, Thomas Everding

OpenGIS® Sensor Event Service Interface Specification (proposed)

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Discussion Paper
Document subtype:	NA
Document stage:	Approved as DP
Document language:	English

Contents

Figures Page

Tables Page

i. Preface

This document contains the specification of a service dedicated to filtering streams of incoming observations based upon user defined filter criteria.

This is a discussion paper; therefore suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

ii. Forward

The contents of this specification may be considered as enhancements for the work on the OGC Sensor Alert Service (SAS). It is not intended to replace SAS, it rather is a suggestion to where work in the area of event stream processing, especially for sensor observation (events) can be heading.

This work was supported by the OSIRIS project. OSIRIS is an Integrated Project (contract number 0033475) co-funded by the Information Society and Media DG of the European Commission within the RTD activities of the Thematic Priority Information Society Technologies.

This document includes three annexes; Annexes A and B are normative and Annex C is informative.

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

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

iii. 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.

iv. Document contributor contact points

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

Name	Organization
Johannes Echterhoff	IFGI
Thomas Everding	IFGI
Jan Torben Heuer	IFGI
Matthes Rieke	IFGI

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2008-08-27	0.0.1	Johannes Echterhoff / Thomas Everding		Initial version
2008-09-29	0.3.0	Carl Reed	Various	Preparation for posting.

vi. Changes to the OGC Abstract Specification

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

vii. Future work

The specification described in this paper is intended to fuel discussion about the following topics that should be considered for future OGC developments:

- Specification of combinable service interfaces rather than standalone service interfaces.

- Server side event stream filtering.

- Usage of OASIS WS-Notification for asynchronous notification of events.

- Definition of policies for communication

- Further elaboration of unit conversion functionality; also, transformation between spatial and temporal reference systems used for incoming event data needs to be addressed

- An investigation and documentation of a precise referencing mechanism for observation properties needs to be performed.

Investigation of sensor data filtering, especially of data streams, has to be pursued in the future. That work will help identifying existing gaps and ultimately lead to robust filter mechanisms on (sensor) data streams.

Introduction

The Sensor Event Service (SES) provides operations to register sensors at the service application and let clients subscribe for observations available at the service. The service performs filtering of sensor data (streams) based upon the filter criteria defined in these subscriptions. Filters can be applied on single observations but also on observation streams, potentially aggregating observations into higher-level information (which itself can be regarded as observation data). Whenever matches are discovered, a notification is sent to the subscriber, using asynchronous, push-based communication mechanisms.

The SES provides functionality for sensor data mining and fusion inside a Spatial Data Infrastructure (SDI). Through its communication model, the SES supports (active) Event Driven Architectures (EDA) and in addition provides mechanisms to support traditional (reactive) Service Oriented Architectures (SOA).

OpenGIS® Sensor Event Service Implementation Specification

1 Scope

This OGC® document specifies an interface for the service side filtering of sensor data (streams) based upon filter criteria defined in subscriptions from clients. The filtering occurs on observations and events that originate from sensors or other data providers (like other event services that are part of a service chain).

The specification defines three different filter levels. Level one filters act upon the syntactical level of sensor data whereas level two and three filters act upon the information contained in sensor data. Level two filters act upon single data tuples while level three filters act upon (combinations of) data streams, allowing the fusion of data into higher-level information. Stream filtering is performed using the filter constructs defined by the Event Pattern Markup Language (EML, see OGC 08-132). In addition, this specification establishes the means for unit conversion of measures used in filter statements.

Observations & Measurements (O&M) is used as sensor data encoding while higher-level information is encoded in an event format compatible with EML.

This specification builds upon the Web Service specifications from the Organization for the Advancement of Structured Information Standards (OASIS). Specifically, this specification reuses the operations and definitions provided by the Web Services Notification (WSN) family of specifications from OASIS. The service defined in this document therefore supports the asynchronous “push” – based communication model.

This document does not impose a restriction to the transport protocols used for performing the specified operations and delivery of notifications. It uses the Web Services Addressing (WSA) specification from W3C to define communication endpoints.

This specification does not define a mechanism for reliable message exchanges between client and service. Such functionality is complementary to the functionality defined in this document and is therefore not in the scope of this specification.

This specification does not cover the functionality for transforming data matching filter criteria of subscriptions into other representations and formats, like the Common Alerting Protocol (CAP).

This OGC® document is applicable to scenarios where on-the-fly and near real-time filtering and aggregation of sensor data is required. It establishes the means for sensor data mining and fusion inside an SDI.

2 Compliance

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

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax (August 1998)*

ISO 19105:2000, *Geographic information — Conformance and Testing*

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

ISO DIS 19136:2006, *Geographic Information — Geography Markup Language*

OASIS WS-BaseNotification, *Web Services Base Notification 1.3 (1 October 2006)*

OASIS WS-BrokeredNotification, *Web Services Brokered Notification 1.3 (1 October 2006)*

OASIS WS-Topics, *Web Services Topics 1.3 (1 October 2006)*

OpenGIS® Best Practices Paper *Definition identifier URNs in OGC namespace OGC Document 06-023r1*

OpenGIS® Best Practices Paper *Geography Markup Language OGC Document 03-105r1*

OpenGIS® Discussion Paper *Event Pattern Markup Language OGC Document 08-132*

OpenGIS® Engineering Specification *Observations and Measurements – Part1: Observation Schema. OGC Document 07-022*

OpenGIS® Implementation Specification *Filter Encoding OGC Document 04-095*

OpenGIS® Implementation Specification *OGC Web Services Common Specification OGC Document 06-121r3*

OpenGIS® Implementation Specification *Sensor Model Language OGC Document 07-000*

UCUM, Unified Code for Units of Measure, Schadow, G. and McDonald, C. J. (eds.), http://aurora.rg.iupui.edu/UCUM

W3C XML Schema Part 1, *XML Schema Part 1: Structures. W3C Recommendation (2 May 2001)*

W3C XML Schema Part 2, *XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001)*

W3C WS-Addressing, *Web Services Addressing 1.0 – Core. W3C Recommendation (9 May 2006)*

W3C XML, *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation (6 October 2000)*

W3C XML Namespaces, *Namespaces in XML. W3C Recommendation (14 January 1999)*

W3C XPath, *XML Path Language 1.0. W3C Recommendation (16 November 1999)*

In addition to this document, this specification includes several normative XML Schema Document files as specified in Annex B.

4 Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

4.1

information broker

relates incoming messages (from information producers) to subscriptions (from information consumers) and delivers matching messages to the notification endpoints contained in subscriptions

4.2

information producer

detects situations of interest and generates messages describing these events

4.3

information consumer

subscriber

subscribes at the information broker for delivery of messages that fulfill given criteria

4.4

filter

defines conditions that are checked against incoming data

NOTE At SES, filtering can be performed on various levels. This may take into account the syntax of incoming messages as well as message content.

4.5

topic

provides a logical grouping of certain message types

EXAMPLES The set of all messages originating from one information producer or the set of messages containing information about a certain event type could each form a topic.

4.6

notification

message

report containing information about one or more events

NOTE In the case of the service defined in this specification, these events are sensor observations and measurements or higher level events.

4.7

event

action that occurs at an instant or over an interval of time [ISO 19136]

NOTE There is another definition given in ISO 19108, where an event is an action that occurs in an instant. An instant is a zero-dimensional geometric primitive representing position in time. More specifically, an instant is an interval whose duration is less than the resolution of the time scale. For this specification, we will follow the definition provided in ISO 19136 (where the event definition that we are using is given in the description of the gml:AbstractTimeSlice element).

4.8
higher-level event
event that contains information derived by processing the information of one or more other events

NOTE Higher-level events represent specialized types of events which may contain the source events from which they have been derived in a causal vector. The information contained in the causal vector allows decision makers to determine why a higher-level event was created (with knowledge of the processing instructions).

4.9
notification endpoint
contains all information required to address and deliver messages to a certain entitiy

4.10
subscription
resource representing the interest of an entity to obtain certain messages (defined by given filter criteria) by delivery of these messages to a given notification endpoint

5 Conventions

5.1 Abbreviated terms

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

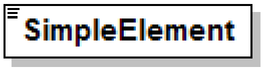
CAP	Common Alerting Protocol
EDA	Event Driven Architecture
EML	Event Pattern Markup Language
O&M	Observations and Measurements
OASIS	Organization for the Advancement of Structured Information Standards
SAS	Sensor Alert Service
SDI	Spatial Data Infrastructure
SES	Sensor Event Service
SOA	Service Oriented Architecture
W3C	World Wide Web Consortium
WSA	Web Services Addressing
WSN	Web Services Notification
XPath	XML Path Language

5.2 XMLSpy 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 normative schema documents perfectly.

5.2.1 Element

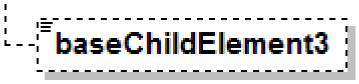
A named rectangle representing the most basic part of the XML Schema notation. Each represents an XML "Element" token. Each Element symbol can be elaborated with extra information as shown in the examples below.



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

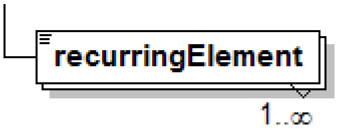
5.2.2 Optional Element

Optional (non mandatory) elements are specified with dashed lines used to frame the rectangle.



5.2.3 Recurring Element

This element (and its child elements if it has any) can occur multiple times.

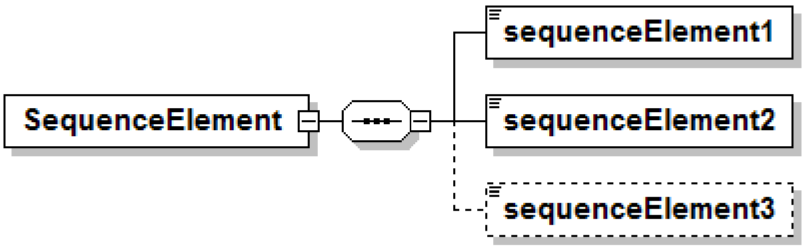


This example shows a recurring element that must occur at least once but can occur an unlimited amount of times. The upper bound here is shown with the infinity symbol.

5.2.4 Sequence Connector

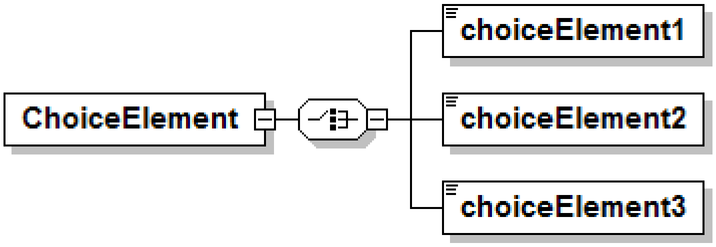
The connection box, called a sequence indicator, indicates that the "SequenceElement" data is made up of three elements. In this example, the first two elements are mandatory and the third element is optional

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



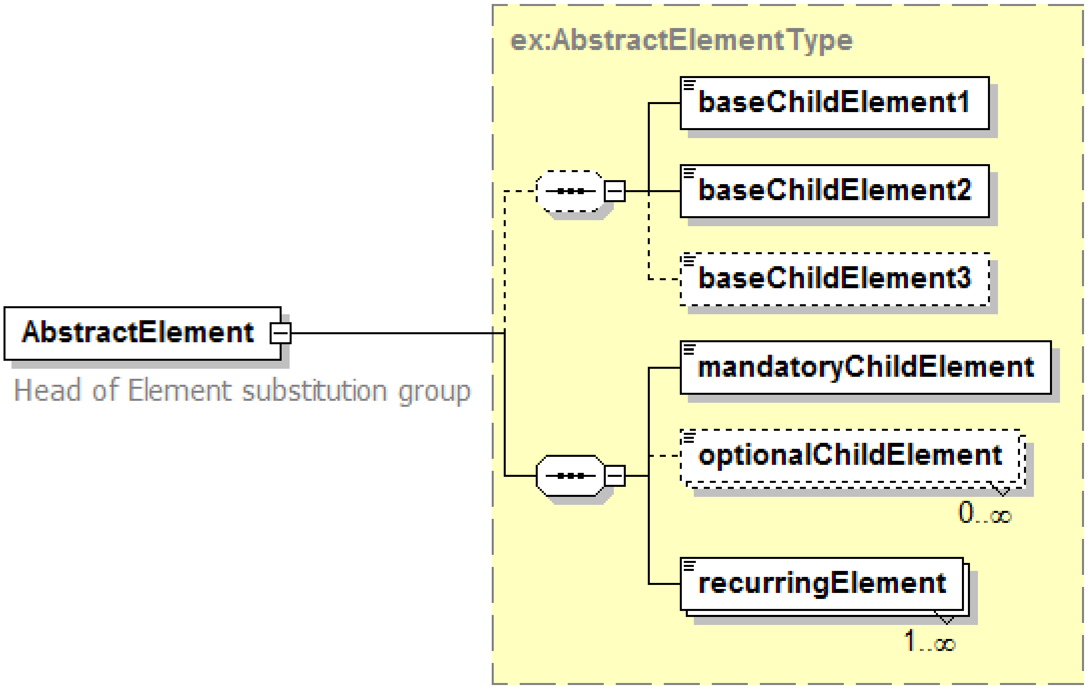
5.2.5 Choice Connector

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.



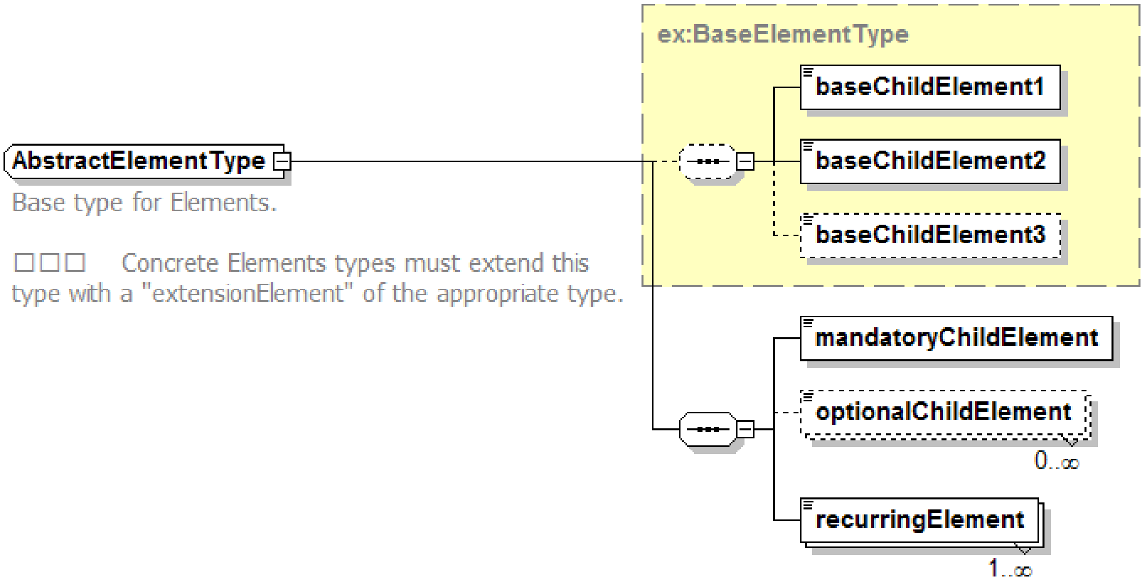
5.2.6 Definition with Complex Type

This diagram illustrates the use of a complex type (i.e., "ex:AbstractElementType") for defining an XML element (e.g., "AbstractElement").



5.2.7 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 can be reused to specify that different elements are of the same type.



6 Sensor Event Service overview

The goal of the Sensor Event Service is to deliver notifications that match the filter criteria defined by subscribers to the notification endpoints defined for a subscription. The service thus acts as an information broker.

NOTE The following information is provided from a rather abstract perspective to establish the conceptual foundation and set of required service functionality.

The service concept can be compared to an event notification system (see Figure 1). Such a system allows information producers to register themselves at the system as well as publish notifications to the system. Information consumers, on the other hand, can subscribe for being notified if events match their subscription criteria.

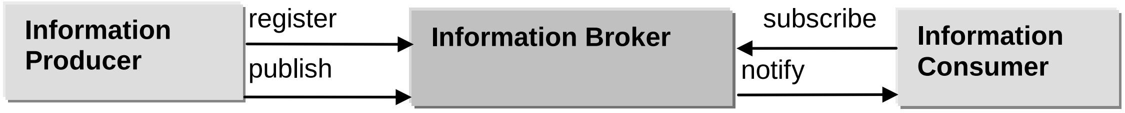


Figure 1: event notification system overview

Three entity roles are involved here: the information producer, broker and consumer. The producer role obviously maps to a sensor while the SES takes the broker role. The consumer role can be any service, application or process - it just needs to provide an endpoint where notifications can be delivered to. Thus, even chaining of information brokers is possible.

Note that only the operations and data types required for handling the conversation between broker and consumer are mandatory in this specification. Therefore, if a provider of the SES wants to leverage an already existing system for gathering notifications from producers – e.g. a SOS connected to multiple sensors, one can simply keep the backside closed and just provide the broker interface for consumers.

In the following, the concepts and further requirements that apply for SES are described.

6.1 Notification

An information broker shall at least be able to handle incoming messages in O&M format. Additional formats for delivering sensor data to the service may be allowed, at the discretion of the implementing service. However, depending on which level of event filtering is supported by the information broker, notifications sent by the service shall either contain one of the data types defined by O&M or the data type defined in this specification for higher-level events (see clause 9.5.2).

6.1.1 Information Producer

This entity is responsible for detecting situations of interest, generating a message describing the event and reporting it. Before being able to actually report an event, the producer has to be registered with at least one information broker. For this, the producer has to provide information about itself and the messages it creates (see clause 6.2). The producer can publish notifications to the broker as long as the registration exists.

6.2 Information Broker

This entity provides interfaces that facilitate the functionality required to register information producers, to receive notifications from them and to subscribe for notifications based upon given filter criteria.

In order to register an information producer, a registration shall provide information about the producer (i.e., the SensorML description of the sensor), the types of messages that the broker should expect to receive from the producer and the topics in which each message should be posted on. Modifications to these types of information may also be published by the information broker.

The broker shall provide operations to create a subscription, to renew it, and also to destroy it. A subscription request shall provide the means to define filter criteria (optional), the desired lifetime of the subscription (optional), whether matching events shall be delivered inside a message container (if such a container is specified and provided by the broker) and the endpoint to where these notifications shall be delivered. It must be specified which topics are covered by a subscription, whether special topic filters are provided in the subscription or not.

6.2.1 Subscription & Registration Resources

Both (consumer) subscription and (producer) registration shall be made available as resources with a well defined lifetime. The lifetime may be finite or infinite depending on the broker. If the lifetime is finite, it may be extended by sending an appropriate request to the broker.

Properties of a resource shall be retrievable through the broker as long as the resource is alive. A resource shall either be destroyed by sending an appropriate request to the broker or automatically once its lifetime is exceeded.

6.2.2 Topics

The information broker shall provide a grouping of incoming notifications into topics. A topic provides a logical grouping of certain message types. The broker shall provide a way to retrieve all available topics. Each SES shall define one topic for all notifications containing O&M data (i.e., sensors observations and measurements) or higher-level events. Additional topics may be defined to provide a more fine grained notification pool at the discretion of the service provider.

6.2.3 Filter Levels

When subscribing, information consumers shall be able to use filter criteria which involve the restriction of messages coming from certain topics only. In addition, filtering shall also appear based upon event content. Three levels of content filtering are defined:

- Level 1 (mandatory): this filter uses XPath to define the filter constraints that shall be checked on each incoming notification separately. At least XPath 1.0 shall be supported.

- Level 2 (optional): this filter uses a combination of logical, spatial, temporal, arithmetic and comparison operators as well as functions to filter new events. Unit of measurement conversion may also be performed. As in level 1 filtering, level 2 filters operate on each incoming event separately.

- Level 3 (optional): in addition to the filter capabilities of level 2, level 3 filters can handle queries on data streams. Thus, they may operate not only on one incoming notification at a time but rather on the whole cloud of events that arrive at the broker.

Which filter capabilities are supported shall be advertised by the broker. At least the first level shall be implemented by each broker. In order to inform a potential subscriber about which filter may be used during subscription, applicable filter types shall be provided for each topic.

6.2.4 Service Metadata

The broker shall provide metadata / capabilities / policies about:

- who is responsible for the service

- which operations are supported

- which filter level is supported; this also involves information about which operators and operands as well as conversion capabilities are supported for filters of level 2 and 3

- whether registration of producers is supported

- how lifetime of registrations is handled

- how lifetime of subscriptions is handled

- whether reliable communication is supported and how it is realized at the service

- which producers are registered at the service

- producer metadata

6.3 Information Consumer (Subscriber)

This entity is the recipient of events matching the criteria defined in a subscription. The consumer does not necessarily have to be the same entity as the one that created the subscription.

7 Notification using WS-* standards

The specifications and standards provided by OASIS and W3C provide a solid foundation for many mainstream IT applications deployed. They offer a comprehensive suite of specifications that provide functionality which can be combined to achieve service specifications that offer the functionality required by the use case and that are interoperable with applications that are also using these standards.

For the Sensor Event Service, the decision was taken to reuse specifications that support publish / subscribe scenarios: the Web Services Notification family of specifications from OASIS. The functionality provided and the extensibility mechanisms supported by WSN are suited for the realization of a Sensor Event Service. Therefore this specification will be based upon the WS-* standards from both OASIS and W3C.

The relevant standards will be listed in the following table. For further information about each of the specifications, please refer to the official documents.

Table 1: WS-* standards relevant for the Sensor Event Service

Title	**Mandatory for SES** **(yes | no)**	**Purpose for the specification**
Web Services Description Language (WSDL)	Yes	The interface, including all operations offered by a certain service instance, is described in a WSDL document. The WSDL document of a given SES will for example identify whether only the mandatory operations (i.e. those offered to information consumers) or also the optional ones are supported.
SOAP	Yes	SOAP defines a framework for exchanging messages between two endpoints. A message container is defined which separates the actual message content from additional information required for example for delivering security relevant information. For SES, SOAP is used as the format for communication.
Web Services Resource (WS-Resource)	Yes	A WS-Resource describes the relationship between a web service and a resource. A WS-Resource offers operations to access, modify and delete a WS-Resource or its properties. For SES, subscriptions of information consumers and registrations of information producers are treated as WS-Resources, for example.

Web Services Notification (WSN)	Yes	WSN allows the registration of notification producers at the service and subsequent publishing of messages to topics supported by the service (for SES, the mandatory topic namespace is described in section 8). Subscriptions can be created which will – if the subscriptions filter criteria are matched by incoming messages – result in the service pushing messages to the notification consumer associated with the subscription. The set of specifications defined by WSN provide the baseline for SES operations.
Web Services Policy (WS-Policy)	No	Policies of a service can be described in its WSDL document. They support the notion of policies that may exist for operations provided by the service, for example the way that reliable messaging is handled by the service instance.
Web Services Reliable Messaging Protocol (WS-ReliableMessaging)	No	Reliable messaging is required in safety critical scenarios. It provides a mechanism to ensure that a message sent to or by a service is delivered once and exactly once.

Other functionality defined by OASIS - for example security mechanisms - and W3C, but even from OGC can also be incorporated into the service interface. This shows that following the modular approach when specifying service functionality is superior compared to defining a monolithic, standalone service.

As the service specified in this document makes use of WS-Notification, it is required to define the topic namespace (see OASIS WS-Topics specification) of the service. The next section will deal with this

8 SES Topic Namespace

The SES shall provide the following mandatory topics. Other topics may be provided if an SES supports the suggested functionality. Topics are defined by the Web Service Topics 1.3 OASIS standard.

8.1 Measurement Topic

The topic *Measurements* is a mandatory topic for all sensor measurements and higher-level events that are provided by the SES. A subscriber can subscribe to all sensors or provide filters to receive only a subset of all incoming sensor data. For this, queries according to one of the three filter levels defined in section 9 can be used.

The message types supported by SES are those from O&M (base specification and extensions) and the SESEvent defined in this specification (see section 9.5.2– SESEvent). In addition, an SES may support other message types specific for the service. In that case, these message types have to be listed in the according topic (or new topics have to be created in the TopicNamespace of the service provider and added to the topic set of the service).

8.2 Sensor Management Topic

The *SensorManagement* topic informs about the current status of sensors. This topic is optional because an SES may only have a fixed set of never-expiring sensors. The supported subtopics are *Registered*, *Unregistered*, *Paused* and *Resumed*. Messages published on this topic shall have the structure shown in Figure 2.

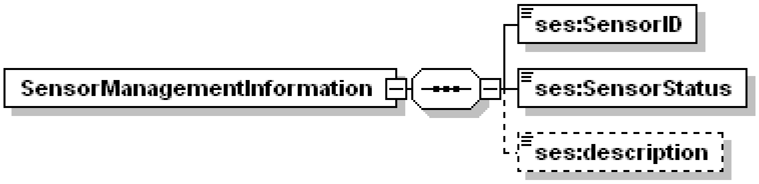


Figure 2: Structure of SensorManagementInformation message element

The parts of the *SensorManagementInformation* message are described further in Table 2.

Table 2: Parts of SensorManagementInformation message

Names	Definition	Data type and values	Multiplicity and use
SensorID	ID of the sensor for which management information is published.	xs:anyURI	one (mandatory)
SensorStatus	Code describing the current status of the sensor. Right now, the only codes defined by SES are: *Registered*: indicates that a sensor with given ID has been registered at the service *Unregistered*: if a sensors registration expired or was deleted *Paused*: if publication of new data by the sensor has been paused *Resumed*: if publication of new data from given sensor has been resumed	xs:token	one (mandatory)
description	Simple string providing a more detailed and human readable description of the management message.	xs:string	zero or one (optional)

8.3 Sensor Expiration Information Topic

This optional topic is used for publishing information about expiration times of sensor registrations. This information gives a hint to subscribers on when a sensor might expire. Note that the expiration time may be extended at any time before the registration expires, this should be expected by all subscribers of this topic. Messages published on this topic shall have the structure shown in Figure 3.

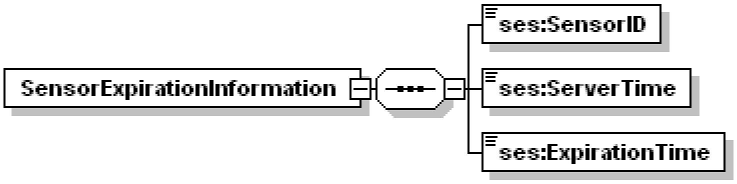


Figure 3: Structure of SensorExpirationInformation message element

The parts of the *SensorExpirationInformation* message are described further in Table 3.

Table 3: Parts of SensorExpirationInformation message

Names	Definition	Data type and values	Multiplicity and use
SensorID	ID of the sensor for which expiration in-formation is published.	xs:anyURI	one (mandatory)
ServerTime	Current time at the service when it generated the message.	xs:dateTime	one (mandatory)
ExpirationTime	Point in time at which the sensor registration will expire. Note that the registration may be renewed after this type of message was created. Therefore this information reflects the state of the service at the given ServerTime.	xs:dateTime	one (mandatory)

8.4 Additional Topics

Other topics, for example used for publishing CAP encoded messages, may be defined by the service provider. Which filter languages apply for these topics is up to the service provider. At least XPath filtering (level 1) shall be supported.

8.5 Topic Namespace Definition

The SES basic TopicNamespace is defined as follows:

Listing 1: SES TopicNamespace

```
<TopicNamespace targetNamespace=”http://www.opengis.net/ses/0.0”
xmlns="http://docs.oasis-open.org/wsn/t-1"
xmlns:om="http://www.opengis.net/om/1.0"
xmlns:omx="http://www.opengis.net/omx/1.0"
xmlns:ses="http://www.opengis.net/ses/0.0">

  <Topic name=”Measurements” messageTypes="om:Observation
omx:Measurement omx:CategoryObservation omx:CountObservation
omx:TruthObservation omx:GeometryObservation
omx:TemporalObservation omx:ComplexObservation ses:SESEvent"/>
      <Topic name=”SensorManagement”
messageTypes=”ses:SensorManagementInformation
ses:SensorExpirationInformation”>
      <Topic name=”Registered”
messageTypes=”ses:SensorManagementInformation" />
      <Topic name=”Deleted”
messageTypes=”ses:SensorManagementInformation" />
      <Topic name=”Paused”
messageTypes=”ses:SensorManagementInformation" />
      <Topic name=”Resumed”
messageTypes=”ses:SensorManagementInformation" />
    </Topic>
    <Topic name=”ExpirationInformation” mes-
sageTypes="ses:SensorExpirationInformation" />
</TopicNamespace>
```

The following example shows a TopicSet of an SES which defines the mandatory topic *Measurements* and the optional topic *ExpirationInformation*. A service provider is free to add additional topics if necessary.

Listing 2: SES TopicSet example

```
<?xml version="1.0" encoding="UTF-8"?>
<wstop:TopicSet xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
xmlns:tses="http://www.opengis.net/ses/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://docs.oasis-open.org/wsn/t-1
http://docs.oasis-open.org/wsn/t-1.xsd"
targetNamespace="http://example.org/yourses">

   <tses:Measurements wstop:topic="true" />
   <tses:ExpirationInformation wstop:topic="true" />
</wstop:TopicSet>
```

9 Filtering Sensor Data

Filtering of sensor data can be divided into two parts. The first way of filtering data is subscribing for a single (or a set of) topic(s) defined in the topic set of the service. This is the easiest approach and provides the subscriber with an aggregation of data having similar or related content. The second way is to define a content-specific filtering. Here the filter is applied to incoming data. The content-specific filtering is realized with three levels, higher ones including lower ones.

Three levels of content filtering are defined:

- Level 1 (mandatory): this filter uses XPath to define the filter criteria that shall be applied on each incoming notification separately. At least XPath 1.0 shall be supported (see example given in clause 9.2).

- Level 2 (optional): this filter uses a combination of logical, spatial, temporal, arithmetic and comparison operators as well as functions. Unit of measure conversion may also be performed. Like level 1 filter operate on single notifications, level 2 filter operate on each incoming event separately.

- Level 3 (optional): in addition to the filter capabilities of level 2, level 3 filters can handle queries on data streams. Thus, they may operate not only on one incoming event at a time but rather on the whole cloud of events that arrive at the broker and in this way make statistical queries (like average or variance of a set of events) possible.

Which filter level and filter functionality is supported by a SES shall be described in its Capabilities document (see section 10.1.1.2).

Before describing the different filter levels in more detail, the unit conversion functionality of a SES needs to be explained in more detail.

9.1 Unit Conversion

Filter statements (e.g. expressed in FES) often contain values of some kind to check whether a given set of data matches a given condition / constraint. These values can be geometries when spatial operations are to be performed, objects representing time when temporal operations are required but also simple scalars for comparison operations. What is important is the reference system (if applicable) these values are given in. Geometries are given in a spatial reference system, times in a temporal one. For scalars, only a subset of data types is of interest with respect to reference systems – these are all quantities that can be associated with a unit of measure (uom).

While it is possible to design service interfaces so that all values in filter statements need to be given in the reference system of the property they should operate on, this is not suitable for SES. This is because SES filters shall be able to operate on data from previously unknown sensors, i.e. on the event cloud generated by all sensors registered at the SES.

Therefore, this specification defines the functionality that allows at least conversion of units of measure to be performed when applying filter statements on incoming data.

To perform unit conversion, a service needs to understand them at least in such terms as to be able to determine that a given uom is compatible with another one. This is the case if uom A can be converted into a set of (SI) base units and uom B can be converted into the same set of base units, meaning that both units are compatible.

As an example, consider the following (weight) units: *British stone* is comparable to *tonne* (symbol: *t*) because they can both be converted into the base unit *gram*.

$$1\ t = 1.000.000\ g$$

$$1\ British\ stone\ = 6.350{,}2932\ g$$

$$\rightarrow\ 1\ t = 157{,}473\ British\ stone$$

This also works for combinations of base units. For example, *Pascal* [unit for pressure; symbol: *Pa*] has the base units *gram*, *meter* and *second*, more specifically:

$$1\ Pa = 1000\ \frac{g}{m^2 \times s^2}$$

The unit *pound per sqare inch* (symbol: *psi*) has the same base units:

$$1\ psi = 6.894.757{,}2932\ \frac{g}{m^2 \times s^2}$$

The two units are therefore comparable and unit conversion can be applied.

This specification focuses on units given by a UCUM code or by a URN. URNs basically define a unit of measure that either has an implied UCUM code (i.e., it can also be converted into a function of base units and is therefore comparable to other units having the same function of base units) or can only be compared to data having the same URN. URNs for units of measure are for example given in OGC 06-023r1 – see Table 4 for example URNs of units of measure and their corresponding UCUM code.

Table 4: List of URNs for Units Of Measure

OGC URN	**EPSG URN**	**UCUM (case sensitive symbol)**
urn:ogc:def:uom:OGC:1.0:degree	urn:ogc:def:uom:EPSG:6.3:9102	deg
urn:ogc:def:uom:OGC:1.0:radian	urn:ogc:def:uom:EPSG:6.3:9101	rad
urn:ogc:def:uom:OGC:1.0:metre	urn:ogc:def:uom:EPSG:6.3:9001	m
urn:ogc:def:uom:OGC:1.0:unity	urn:ogc:def:uom:EPSG:6.3:8805	1

Additional URNs may be used in applications. However, usage of UCUM codes is recommended. Which units of measure a service recognizes and understands (i.e., the service knows to which units a given unit can be compared to) shall be indicated in the services Capabilities document (see clauses 9.4 and 10.1.1.2).

If a service is not capable of performing unit conversion, it shall ignore all units of measure, i.e. wherever a comparison of properties occurs, the service shall simply compare the property value with the given value.

9.2 Level 1 Filter

XPath 1.0 is the mandatory filter language for SES message content filters. XPath 2.0 filter statements can also be supported by a specific service instance.

An XPath statement is valid if it returns a boolean result, so either true or false. A boolean result is required so that the service can decide whether a message matches the filter or not.

NOTE: Filtering via XPath is highly depending on the syntax of observation instance documents. In addition, one never knows whether observations provide elements inline or by reference. So care needs to be taken when formulating XPath statements.

Describing the details of XPath based filtering is out of scope for this specification. XPath is the topic of numerous tutorials and books, so the reader is referred to these sources for further information about XPath. Nevertheless, we provide one example with several XPath filter statements and their outcome when applied to a given O&M instance document.

Example

Given the following xml-document, encoded in O&M:

Listing 3: O&M example for level 1 filter queries

```
<?xml version="1.0" encoding="UTF-8"?>
<om:Observation (…)>
  (…)
  <om:procedure xlink:href=
             "urn:ogc:object:procedure:CITE:WeatherService:LGA"/>
  <om:observedProperty>
    <swe:CompositePhenomenon gml:id="cpid0" dimension="2">
      <gml:name>resultComponents</gml:name>
      <swe:component xlink:href="urn:ogc:data:time:iso8601"/>
      <swe:component xlink:href=
                     "urn:ogc:def:property:OGC:1.0:temperature"/>
    </swe:CompositePhenomenon>
  </om:observedProperty>
  <om:featureOfInterest>
    <gml:FeatureCollection>
      <gml:featureMember>
        <sa:SamplingPoint gml:id="FOI_LGA">
          <gml:name>La Guardia Airport</gml:name>
          <sa:sampledFeature xlink:href=
                     "http://my.service.org/if?getFeature=F_01"/>
            <sa:position>
              <gml:Point>
                <gml:pos srsName="urn:ogc:crs:epsg:4326">
                    40.7772 -73.8725
                </gml:pos>
              </gml:Point>
            </sa:position>
          </sa:SamplingPoint>
        </gml:featureMember>
      </gml:FeatureCollection>
    </om:featureOfInterest>
  <om:result>
   (…)
  </om:result>
</om:Observation>
```

The following example of an XPath expression illustrates how a reasonable XPath-filter query could look like. With the expression

```
//om:Observation/om:procedure/@xlink:href='urn:ogc:object:procedu
re:CITE:WeatherService:LGA'
```

a boolean-test is evaluated. It returns true if there is a procedure with xlink:href attribute value of `urn:ogc:object:procedure:CITE:WeatherService:LGA`. This shows how to easily subscribe for messages from only a single sensor. If you know exactly that the identifier of the procedure / sensor is a specific URN and that this URN (as should be implied by the URN concept) is unique, then you can also query directly for the containment of this identifier with the following XPath expression:

```
//@xlink:href='urn:ogc:object:procedure:CITE:WeatherService:LGA'
```

Another XPath-expression could be:

```
//om:Observation//om:observedProperty//@xlink:href='urn:ogc:def:p
roperty:OGC:1.0:temperature'
```

With this filter query, one would subscribe for all observations containing the observed property `urn:ogc:def:property:OGC:1.0:temperature`. This example again shows that the syntax of the given XML instance is

Another XPath example could be to combine the previous two expressions with a simple 'and' operator. This would result in subscribing for observations containing both the given station as well as the property temperature:

```
//om:Observation/om:procedure/@xlink:href='urn:ogc:object:procedu
re:CITE:WeatherService:LGA' and
//om:observedProperty//swe:component/@xlink:href='urn:ogc:def:pro
perty:OGC:1.0:temperature'
```

The following XPath expression would return false because there is a typo:

```
//om:Observaton/om:procedure/@xlink:href='urn:ogc:object:procedur
e:CITE:WeatherService:LGA'
```

9.3 Level 2 Filter

The second level, as opposed to the first, operates on the actual content of (i.e., the information contained in) the measured data rather than on the syntactic level alone. Remember that level two filters for SES are only defined for the *Measurement* topic.

Filtering with value-thresholds, spatial operations and time-based filters can be performed. One simple example is filtering for

„urn:ogc:def:property:OGC:1.0:temperature“ measured greater or equal than 29° C.

In this filter example a conversion to the base units (see clause 9.1) of the property should happen if the service receives a temperature reading expressed in *°F* and if this functionality is supported by the service (see clause 9.4.2).

The SES thus allows the comparison of sensor data even if the filter value and the data have different but compatible units of measure.

9.4 Filter_Capabilities

Which functions an SES instance supports shall be declared in its *Filter_Capabilities*. Filter_Capabilities consist of the following six main capability components. If a component is not supported by the service the according element shall be omitted.

NOTE The filter functionality of SES heavily relies upon and extends the functionality defined by the Filter Encoding Specification (FES).

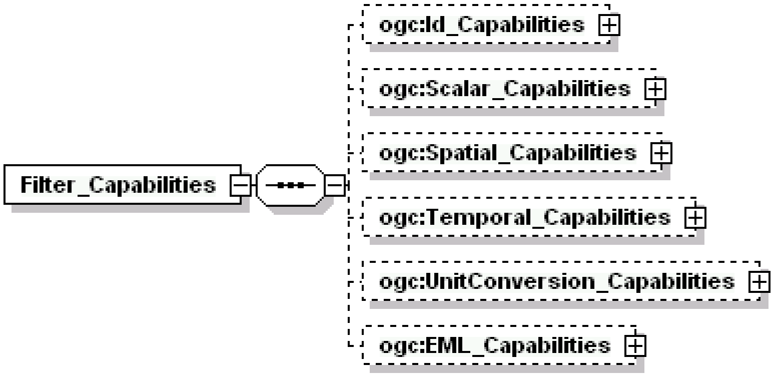


Figure 4: Filter capabilities sections of the SES service

The *Id_Capabilities*, *Spatial_Capabilities* and *Scalar_Capabilities* are realized according to the FES. The other elements have either been added or imported to accomplish the SES functionality.

9.4.1 Temporal Capabilities

Temporal filtering is not yet part of an official OGC Filter Encoding specification, but has been specified in past OGC testbeds and incorporated e.g. into the Sensor Observation Service Specification ([1]) as an extension to the existing filter encoding. In addition, work is underway to provide an updated version of the Filter Encoding specification to ISO (which will become ISO 19143). The SOS filter encoding extension and the existing ISO draft differ only in naming of attributes and operators of the temporal filter. The functionality remains the same. Therefore, we will use the operators and elements defined in the ISO draft.

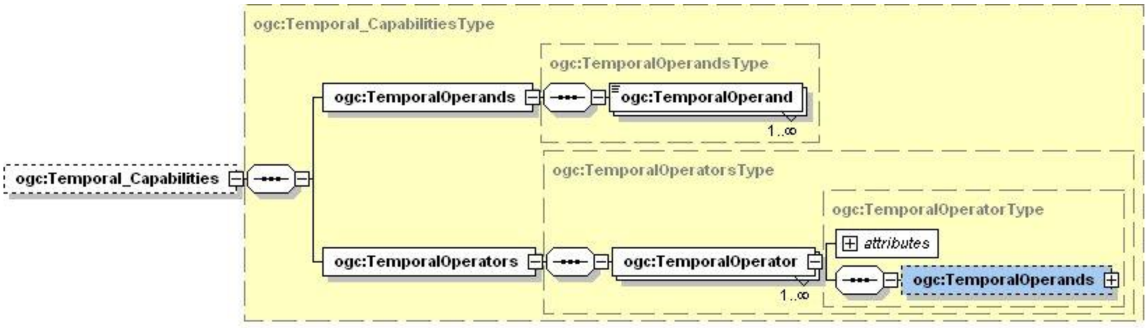


Figure 5: structure of the temporal capabilities section

The temporal capabilities define which temporal operators and operands are supported by a specific service instance. A temporal operator determines whether its time arguments satisfy the stated temporal relationship. The operator evaluates to true if the temporal relationship is satisfied. Otherwise the operator evaluates to false.

The service shall declare in its filter capabilities which temporal operators it supports. Temporal operators are used to test whether the value of a temporal property, referenced using the name of the property, and a literal temporal value satisfy the temporal relationship implied by the operator. Temporal values are expressed using GML (see OGC 03-105r1).

The semantics of the temporal operators are defined in clause 5.2.3.5 of ISO 19108. However, Table 5 lists all defined temporal operations together with their corresponding ISO 19108 enumeration value.

Table 5: Mapping of temporal filter operators to ISO 19108 relative position result values

temporal filter operator	ISO 19108 temporal operator
After	After
Before	Before
Begins	Begins
BegunBy	Begun
TContains	Contains
During	During
TEquals	Equals
TOverlaps	Overlaps
Meets	Meets
OverlappedBy	OverlappedBy
MetBy	MetBy
EndedBy	EndedBy

9.4.2 Unit Conversion Capabilities

Support for conversion of units of measurement (see clause 9.1) shall be declared in the SES Capabilities. There are two ways of supporting the conversion of units.

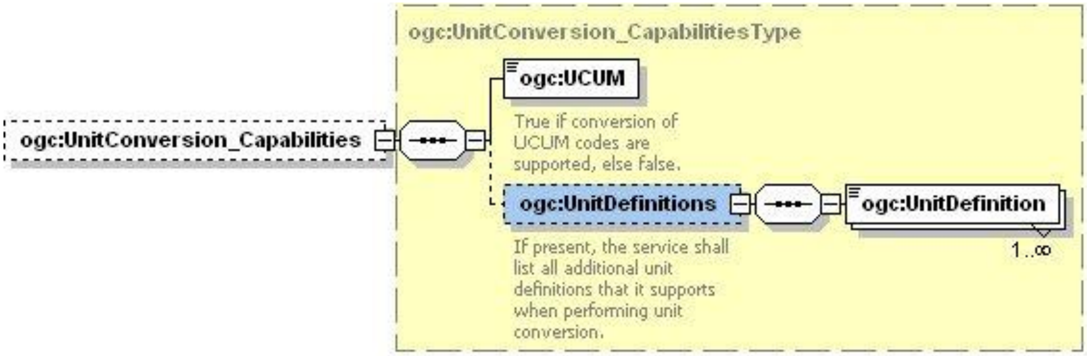

ogc:UnitConversion_CapabilitiesType
ogc:UnitConversion_Capabilities
ogc:UCUM
True if conversion of UCUM codes are supported, else false.
ogc:UnitDefinitions
If present, the service shall list all additional unit definitions that it supports when performing unit conversion.
ogc:UnitDefinition
1..∞

Figure 6: Components of the unit conversion capabilities element

The <ogc:UCUM> element in the UnitConversion_Capabilities defines whether the service supports conversion of UCUM codes or not. If set to true the service shall support conversion of all UCUM units. In addition, the service can declare support for specific unit definitions in the <ogc:UnitDefinitions> element by listing their assigned URNs.

The OGC Filter Encoding does not specify that literals used in comparison filters shall provide their unit of measure if the referenced property is a quantity. For the SES, literals used in comparison operators shall use one of the element types defined in the gml:ScalarValue group (see Figure 7).

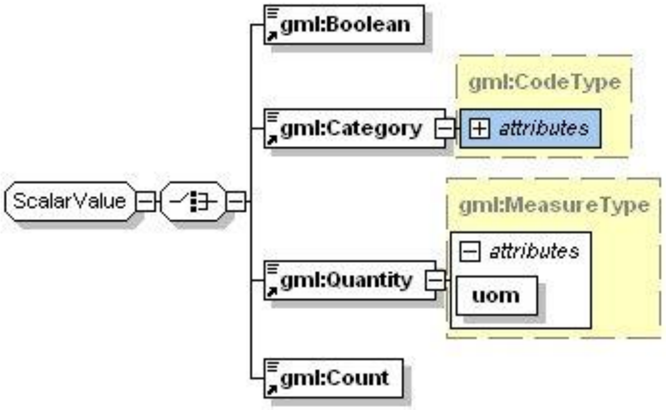


Figure 7: allowed content of the ogc:Literal element in expressions of comparison filters

Whenever the property referenced in the operator is a quantity, the gml:Quantity should be used to provide the value to compare the property value against – as well as the unit of measure (also UCUM codes in addition of URNs for providing the uom are allowed) in which the comparison value is expressed. Likely, gml:Boolean, gml:Category or gml:Count should be used if the referenced property is expected to be a boolean, category or count.

9.4.3 Filter Statements

When subscribing at SES, clients can provide filter statements to further define in which messages they are interested in. These filter statements operate on the content of messages. A filter statement is created by providing one or a combination of the components shown in Figure 8.

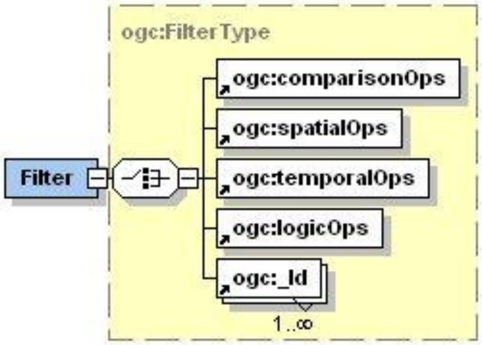


Figure 8: filter element showing the five different filter types that can be specified in queries

As said before, comparison operations require one of the gml:ScalarValue group elements as content of the literal used in the filter expression. This enables the service to easily determine which data type is given in a filter criterium and in addition also which unit of measure is provided for a given value (if a quantity is used).

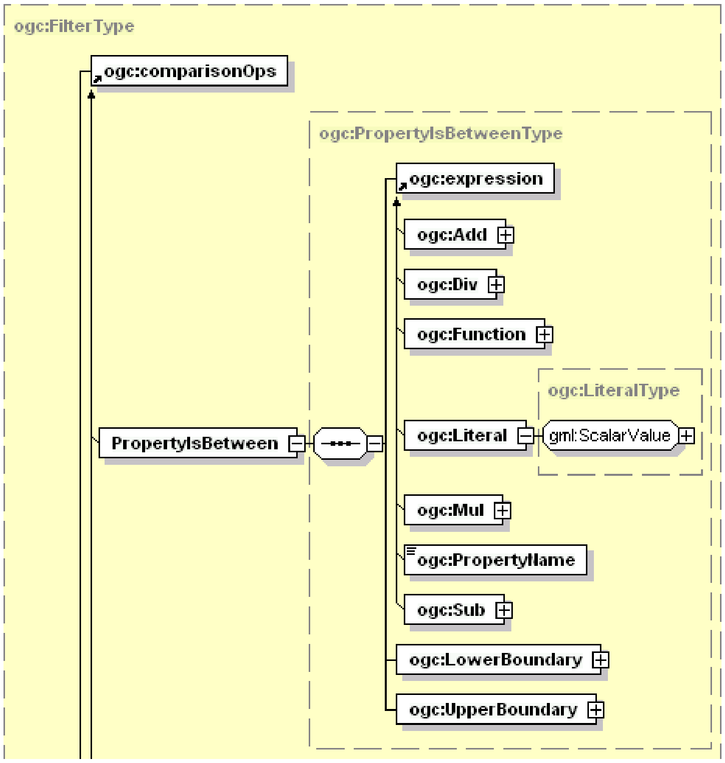


Figure 9: filter schema excerpt showing the allowed content of a literal in comparisonOps: one of the gml:ScalarValue element group

The comparison, logical, spatial and id operators work as defined by the OGC Filter Encoding specification. Temporal operations are not yet part of the current specification (OGC 04-095). They have thus been included to the list of available filter operators for this service.

Example

Following is an example of a reasonable (level 2) filter statement with three defined conditions. The application of this filter will be shown using the following observation (of a sensor measuring temperature).

Listing 4: level 2 filter example – observation document

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<om:Observation (…)>
  <om:samplingTime>
    <gml:TimeInstant gml:id="ot1t">
      <gml:timePosition>
            2008-06-01T07:45:00+01Z
         </gml:timePosition>
    </gml:TimeInstant>
  </om:samplingTime>
  <om:procedure xlink:href=
                         "urn:ogc:object:procedure:IfGI:S1324"/>
  <om:observedProperty xlink:href=
                    "urn:ogc:def:property:OGC:1.0:temperature"/>
  <om:featureOfInterest>
    <sa:SamplingPoint>
      <gml:name>Beyenburg</gml:name>
      <sa:sampledFeature xlink:href=
                    "http://my.service.org/if?getFeature=F_01"/>
      <sa:position>
        <gml:Point gml:id="p1">
          <gml:pos srsName="urn:ogc:def:crs:EPSG:6.17:31466">
            2590759 5680179
          </gml:pos>
        </gml:Point>
      </sa:position>
    </sa:SamplingPoint>
  </om:featureOfInterest>
  <om:result xsi:type="swe:QuantityPropertyType">
    <swe:Quantity definition=
                      "urn:ogc:def:property:OGC:1.0:temperature">
      <swe:uom code="[degF]"/>
      <swe:value>102.9</swe:value>
    </swe:Quantity>
  </om:result>
</om:Observation>
```

The following filter would evaluate to true when applied to the given observation:

Listing 5: level 2 filter example – filter statement

```
<?xml version="1.0" encoding="UTF-8"?>
<Filter (…)>
  <And>
    <PropertyIsGreaterThan>
      <PropertyName>
                  urn:ogc:def:property:OGC:1.0:temperature
      </PropertyName>
      <Literal>
        <gml:Quantity uom="Cel">30</gml:Quantity>
      </Literal>
    </PropertyIsGreaterThan>
    <Contains>
      <PropertyName>geometry</PropertyName>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.17:31466">
        <gml:lowerCorner>2590000 5680000</gml:lowerCorner>
        <gml:upperCorner>2590800 5680200</gml:upperCorner>
      </gml:Envelope>
    </Contains>
    <After>
      <PropertyName>startTime</PropertyName>
      <gml:TimeInstant>
        <gml:timePosition>2008-05-25T12:25:00Z</gml:timePosition>
      </gml:TimeInstant>
    </After>
  </And>
</Filter>
```

The statement combines both a comparison expression with a spatial and temporal expression with a logical AND.

The first filter component applies a constraint to the measured property *urn:ogc:def:property:OGC:1.0:temperature*, evaluating to true if the measured value is greater than 30 °C. The unit of measure ("Cel" - the UCUM code for degree Celsius) is defined in the <Literal> element using a <gml:Quantity>. If supported by the service, unit conversion will be performed so that the measure given in degree Fahrenheit and the filter value given in degree Celsius will be transformed into their base unit (Kelvin) before applying the comparison filter.

Second is a simple spatial filter evaluating to true if the geometry of the observation is within the given <gml:Envelope>. For SES, the keyword *geometry* shall be used whenever a filter shall be applied to the geometric property of an event.

The last component applies a filter on the (sampling) time of the observation. A sampling time after the given <gml:TimeInstant> resulting in a match when the sampling time of the observation is after 2008-05-25T12:25:00Z. For SES, the keywords *startTime* & *endTime* shall be used whenever a filter shall be applied to the temporal property of an event.

When filtering for an observation referencing a sensor with the givenID, the keyword *sensorID* shall be used.

9.5 Level 3 Filter

The highest level of filtering functionality at SES is level three. Here, content based filters acting on (the combination of) event streams rather than on single events (as in level two) are available. Time windows as a specialized view on which events can be part of a filter query are available, for example. Filtering will then include all events that have been received by the service inside the given time window.

Simple examples for stream filter use cases are:

- the counting of events that occurred inside a given time window
- calculating the average of a measured phenomenon over a certain time period
- being informed only when a threshold is crossed
- count of new events with certain property which exceeds the given threshold inside a given time window
- average of property from the last five events from sensor XYZ is greater than a given value

It is also possible to correlate the inputs of multiple event streams / sensors. This type of filtering produces results derived from more than one sensor, the so called higher-level events. Such results are posted as a SESEvent (see clause 9.5.2). They contain correlated values and a causal vector of the events that led to the higher-level event. In order to enable cascading of multiple SES it is also possible to receive SESEvents as input for filtering.

This specification uses the Event Pattern Markup Language (EML) to define level 3 filter statements. EML uses FES filter statements in guards that define constraints on events – this means that level two is in fact a subset of filter level three. The concepts and constructs of EML will not be repeated in this specification. Only the specific requirements of SES with respect to the definition and application of EML patterns shall be specified in the following.

- definition of input name in simple patterns
- definition of specialized event object
- keywords for additional generic event properties
- handling of incoming data
- handling output(s)
- handling of notify on select

9.5.1 InputName in EML SimplePattern

For SES, the InputName attribute of an EML SimplePattern specifies which sensor is used as the data source for the pattern. This means that a subscriber can either provide a specific sensor ID for the InputName of a SimplePattern (meaning that the pattern only looks for events from that specific sensor) or the wildcard '*' to reference all sensors registered at SES (as long as the subscription lasts).

9.5.2 SESEvent

EML specifies a generic event data type which should be incorporated by specifications using EML. The SESEvent (see Figure 10) is used by SES as data structure to encode the information of higher-level events as well as to send and receive them.

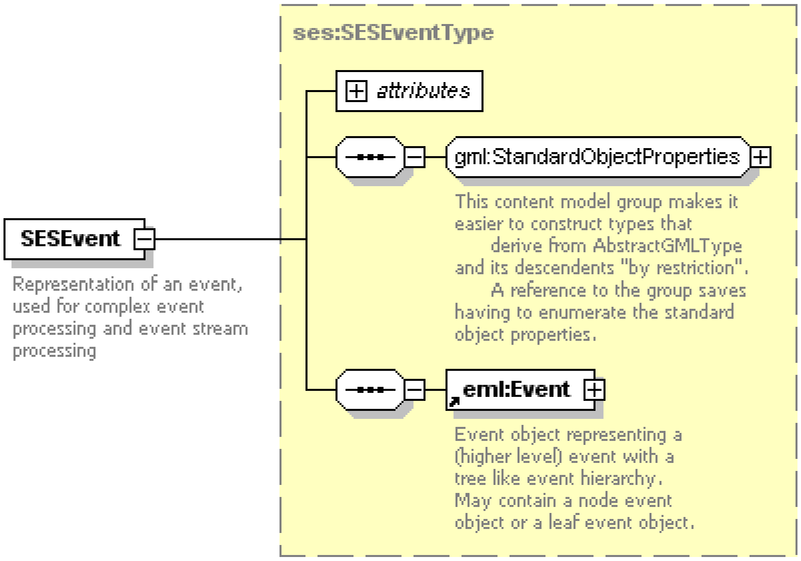

ses:SESEventType
attributes
gml:StandardObjectProperties
This content model group makes it easier to construct types that derive from AbstractGMLType and its descendents "by restriction". A reference to the group saves having to enumerate the standard object properties.
SESEvent
Representation of an event, used for complex event processing and event stream processing
eml:Event
Event object representing a (higher level) event with a tree like event hierarchy. May contain a node event object or a leaf event object.

Figure 10: Structure of the SESEvent type

The SESEventType extends gml:AbstractGMLType. Therefore, a SESEvent is in the substitution group of gml:_GML. The eml:Event structure is simply embedded inside a SESEvent. Only the (optional) attributes and standard object properties from gml:AbstractGMLType are added.

The leaf attribute of the eml:Event shall only contain O&M Observations (when that observation is contained in the causal vector of an event) or eml:Events (in cases where the first causing event was received as higher-level event from another SES).

In addition to the generic properties of the eml:Event, a SESEvent shall contain an identifier for the process that generated the event (usually the newEventName assigned by SelectFunctions) and a geometry attribute. The geometry of a higher level SESEvent is stored as a property named *geometry*. If this property does not exist the geometry is defined by default as the union of all geometries defined by the contained SESEvents.

9.5.3 Keywords for Generic Event Properties

EML defines rules for how event properties shall be accessed (e.g. in guards, views and select functions). In addition to the *value*, *startTime*, *endTime*, *causality* and '*' keywords defined by EML for accessing the generic properties of events, this specification adds the keywords *sensorID* for accessing the sensor ID or the newEventName (assigned to higher-level events) and *geometry* for accessing the geometry.

9.5.4 Handling of Incoming Data

The source of an event can either be a sensor or another SES. This means that a SES shall expect to receive O&M Observation or ObservationCollection instance documents or SESEvents. An ObservationCollection shall be split up by the service. Each Observation shall be transformed into one event object unless the Observation contains multiple datasets (for instance by using a SweCommon DataArray) instead of a single data tupel. In such a case an event object is created for every contained tupel.

9.5.5 EML Output Handling

In EML, the assignment of filter processing results is performed via the outputName attribute in SelectFunction(s). The keyword to be used as output name whenever a selection shall be sent to the subscriber is simply *Output*. If the outputName is missing or empty the result of the select function will only be available internally to further patterns in the EML processing chain.

An important aspect for SES is how to deliver results to subscribers. Whenever an event only has a single observation in its causal vector or only stores one observation in its content, the observation will be sent to the client. In all other cases a SESEvent will be delivered.

9.5.6 Notify On Select

The functionality of the NotifyOnSelect function in EML is not sophisticated enough to be used by SES at this time. It shall therefore be ignored by SES.

10 SES Operations

The operations defined hereafter apply to two different port types (for further information on port types see the WSDL specification). Port types resemble interfaces which can be combined in a single web services interface definition. The following table provides an overview of which port types shall provide which operations. Whether a certain port type is optional or mandatory is shown as well.

Table 6: Overview of the PortTypes relevant for an SES implementation together with associated operations

PortType name	Base definition from which WS-* specification	PortType mandatory for SES service? (Yes | No)	Operation
bSES (basic SES)	Extends NotificationProducer defined in WS-BaseNotification	Yes	GetCapabilities Describe Sensor Subscribe GetCurrentMessage
SubscriptionManager	SubscriptionManager in WS-BaseNotification	Yes	Renew Unsubscribe
PausableSubscription Manager	PausableSubscription Manager in WS-BaseNotification	No	PauseSubscription ResumeSubscription
PullPoint	PullPoint in WS-BaseNotification	No	CreatePullPoint GetMessages DestroyPullPoint
eSES (enhanced SES)	extends bSES PortType defined in this specification	No	Notify RegisterPublisher
SesPublisher RegistrationManager	Extends PublisherRegistration Manager from WS-BrokeredNotification	No	RenewRegistration DestroyRegistration

In addition to the operations defined for each port type, an implementation may also support the required and optional message exchanges defined in the WS-ResourceProperties specification. These operations are listed in Table 7. If an implementation chooses to implement the operations of WS-ResourceProperties and WS-ResourceLifetime (see Table 8), then the mandatory resource properties for the resulting resource types shall be provided in accordance with the specifications of WS-Notification. Where a new port type is defined by this specification by extending those from WS-Notification, additional resource properties may become mandatory.

Table 7: Overview of the operations defined by WS-ResourceProperties

WS-ResourceProperties Operation	**Mandatory (Yes | No)**
GetResourceProperty	Yes
GetResourcePropertyDocument	No
GetMultipleResourceProperties	No
QueryResourceProperties	No
PutResourcePropertyDocument	No
SetResourceProperties	No
InsertResourceProperties	No
UpdateResourceProperties	No
DeleteResourceProperties	No

Table 8: Overview of the operations defined by WS-ResourceLifetime

WS-ResourceLifetime Operation	**Mandatory (Yes | No)**
Destroy	Yes
SetTerminationTime	Yes

Only if the realization of operations defined for the port types listed in Table 6 differs from those provided in their base specifications or if they are new operations defined for SES will they be described in more detail in this document. Otherwise, the reader is referred to the respective documents.

10.1 Basic SES Port Type

The basic SES port type combines the GetCapabilities, DescribeSensor, Subscribe and GetCurrentMessage operations.

10.1.1 GetCapabilities operation

The GetCapabilities operation is used to retrieve general metadata about the service. This includes all sections defined by OWSCommon as well as SES specific content sections. Via the GetCapabilities operation a client can determine which sensors are currently registered as information publishers as well as which filter capabilities the service provides.

10.1.1.1 Request

The operation is invoked by sending a GetCapabilities request to the service.

The WS-Addressing (action) Message Addressing Property shall contain the URI:
http://www.opengis.net/ses/GetCapabilitiesRequest

Conforming to OGC practice, a request might look like the following:

Listing 6: GetCapabilities request example

```
<?xml version="1.0" encoding="UTF-8"?>
<ses:GetCapabilities service="SES"
xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

10.1.1.2 Response

The response to a valid GetCapabilities request shall be a Capabilities document containing the sections requested. The Contents section lists the IDs of all sensors currently registered as information publishers while the Filter_Capabilities section (see section 9.4) provides information about which filter functionality is supported by the respective service instance.

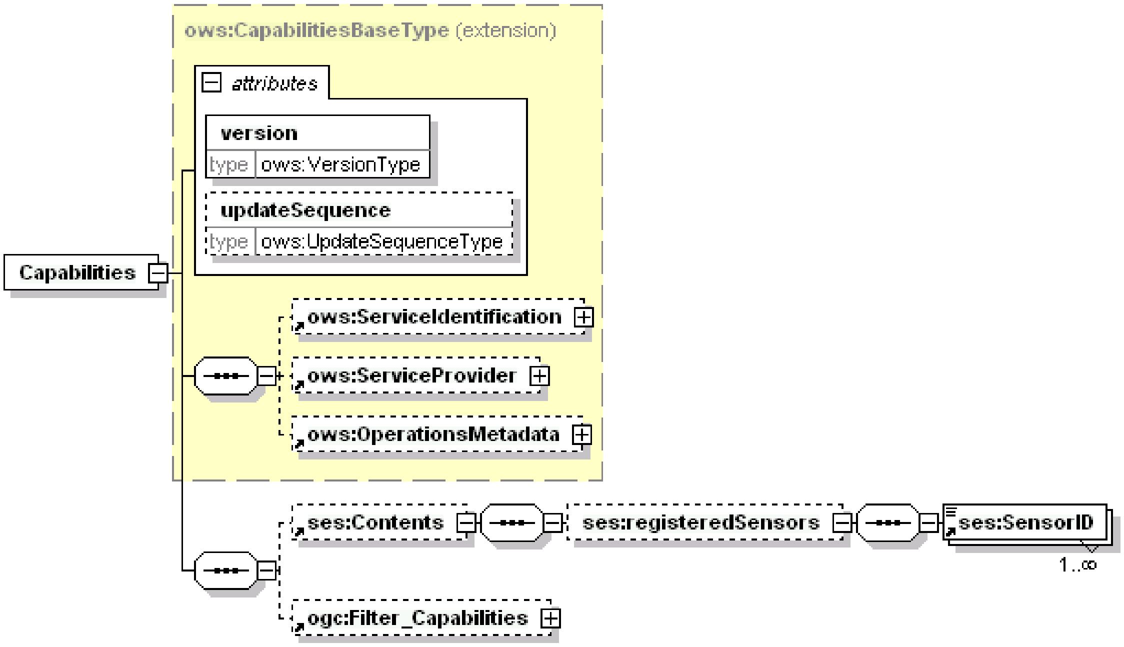


Figure 11: Structure of the SES Capabilities element

The WS-Addressing (action) Message Addressing Property shall contain the URI: http://www.opengis.net/ses/GetCapabilitiesResponse

10.1.2 DescribeSensor operation

The DescribeSensor operation requests the SensorML definition of the given Sensor. The response is the SensorML document for the sensor.

10.1.2.1 Request

The structure of the DescribeSensor operation request shall be as follows:

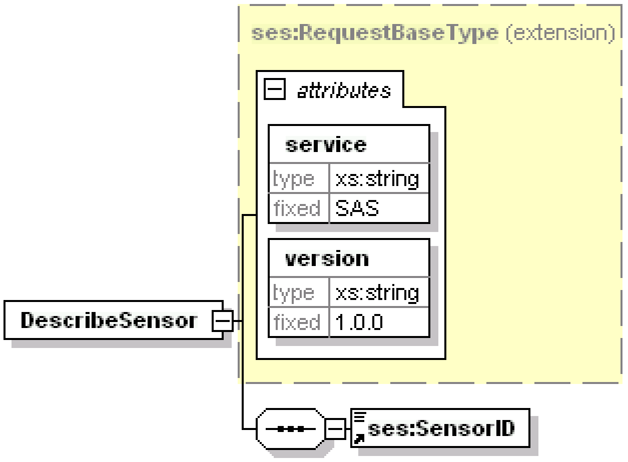


Figure 12: Structure of the DescribeSensor request element

The WS-Addressing (action) Message Addressing Property shall contain the URI: http://www.opengis.net/ses/DescribeSensorRequest

An example request could look like the one shown in the following listing.

Listing 7: DescribeSensor operation request example

```
<?xml version="1.0" encoding="UTF-8"?>
<ses:DescribeSensor service="SES" version="1.0.0"
xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<ses:SensorID>
   urn:ogc:object:sensor:IFGI:1.0:Temp:1</ses:SensorID>
</ses:DescribeSensor>
```

10.1.2.2 Response

The response to a valid DescribeSensor request is the SensorML description of the referenced sensor.

The WS-Addressing (action) Message Addressing Property shall contain the URI: http://www.opengis.net/ses/DescribeSensorResponse

The example response shown in Listing 8 is taken from the SensorML specification. You can find more detailed information in the SensorML specification.

Listing 8: DescribeSensor operation response example

```
<?xml version="1.0"?>
<sml:SensorML xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
   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"
   version="1.0">
   <member xlink:role="urn:ogc:def:role:OGC::sensorSystem">
        <System gml:id=" urn:ogc:object:sensor:IFGI:1.0:Temp:1">
            <gml:description> ... </gml:description>
            <gml:name> ... </gml:name>
        ...
    </System>
   </member>
</sml:SensorML>
```

10.1.3 Subscribe operation

The subscribe operation is adopted from the WS-BaseNotification specification and shall be used as specified in that document. Because this operation is among the most relevant for this specification, it will be described in more detail.

10.1.3.1 Request

The request structure for this operation is defined in the WS-BaseNotification specification.

The WS-Addressing (action) Message Addressing Property shall – as specified in WS-BaseNotification - contain the URI:
http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeRequest

Following is a simple example of a Subscribe request. The filtering in this example is limited to topic- and level-1-filtering.

Listing 9: Subscribe operation request example

```
<?xml version="1.0" encoding="UTF-8"?>
<wsnt:Subscribe xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:tses="http://www.opengis.net/ses/1.0.0"
xmlns:om="http://www.opengis.net/om/1.0"
xmlns:xlink="http://www.w3.org/1999/xlink">
<wsnt:ConsumerReference>
     <wsa:EndpointReferenceType />
       <--! as defined in WS-Addr-Specification} -->
     </wsnt:ConsumerReference>
     <wsnt:Filter>
       <wsnt:TopicExpression Dialect=
     "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
          tses:Measurements
       </wsnt:TopicExpression>
       <wsnt:MessageContent Dialect=
          "http://www.w3.org/TR/1999/REC-xpath-19991116">
//@xlink:href='urn:ogc:object:procedure:CITE:WeatherService:LGA'
       </wsnt:MessageContent>
     </wsnt:Filter>
     <wsnt:InitialTerminationTime>
          2008-06-25T12:00:00.00000Z
     </wsnt:InitialTerminationTime>
```

```
</wsnt:Subscribe>
```

10.1.3.2 Response

The response structure for this operation shall be as defined in the WS-BaseNotification specification.

The WS-Addressing (action) Message Addressing Property shall – as specified in WS-BaseNotification - contain the URI:
http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeResponse

Following is a simple example response to the request provided in the previous clause.

Listing 10: Subscribe operation response example

```
<?xml version="1.0" encoding="UTF-8"?>
<wsnt:SubscribeResponse
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsnt:SubscriptionReference>
        <wsa:Address>
            http://www.example.org/SubscriptionManager-id-1
        </wsa:Address>
    </wsnt:SubscriptionReference>
</wsnt:SubscribeResponse>
```

10.1.4 GetCurrentMessage operation

The GetCurrentMessage operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.2 Subscription Manager Port Type

The SubscriptionManager port type combines the Renew and Unsubscribe operations.

10.2.1 Renew operation

The Renew operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.2.2 Unsubscribe operation

The Unsubscribe operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.3 Pausable Subscription Manager Port Type

The PausableSubscriptionManager port type combines the PauseSubscription and ResumeSubscription operations.

10.3.1 PauseSubscription operation

The PauseSubscription operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.3.2 ResumeSubscription operation

The ResumeSubscription operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.4 Pull Point Port Type

The PullPoint port type combines the CreatePullPoint, GetMessages and DestroyPullPoint operations.

10.4.1 CreatePullPoint operation

The CreatePullPoint operation shall be implemented and used as defined in the WS-BaseNotification.

10.4.2 GetMessages operation

The GetMessages operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.4.3 DestroyPullPoint operation

The DestroyPullPoint operation shall be implemented and used as defined in the WS-BaseNotification specification.

10.5 Enhanced SES Port Type

The enhanced SES port type combines the Notify and RegisterPublisher operations.

10.5.1 Notify operation

The Notify operation shall be implemented and used as defined in the WS-BaseNotification specification.

A Notify request could look like the following:

Listing 11: Notify request containing an Observation

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://www.w3.org/2003/05/soap-envelope"

xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ses="http://www.opengis.net/ses/1.0.0"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
```

```
xmlns:om="http://www.opengis.net/om/1.0"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <wsa:Action>
  http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
    </wsa:Action>
    <!-- ... -->
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:Notify>
      <wsnt:NotificationMessage>
        <wsnt:SubscriptionReference>
          <wsa:Address>
            http://www.example.org/SubscriptionManager
          </wsa:Address>
        </wsnt:SubscriptionReference>
        <wsnt:Topic Dialect=
     "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
          ses:Measurement
        </wsnt:Topic>
        <wsnt:ProducerReference>
          <wsa:Address>
            http://www.example.org/NotificationProducer
          </wsa:Address>
        </wsnt:ProducerReference>
        <wsnt:Message>
          <om:Observation>
            <om:samplingTime>
              <gml:TimeInstant gml:id="ot1t">
                <gml:timePosition>
                    2005-01-11T16:22:25.00
                </gml:timePosition>
              </gml:TimeInstant>
            </om:samplingTime>
            <om:procedure xlink:href=
       "urn:ogc:def:feature:sensor:OGC:1.0:IFGI:SID120u7381381"/>
            <om:observedProperty xlink:href=
                "urn:ogc:def:property:OGC:1.0:mass"/>
            <om:featureOfInterest xlink:href=
                "http://my.wfs.org?getF=asd7897asd6"/>
            <om:parameter>
              <swe:Quantity definition=
                "urn:ogc:def:property:OGC:1.0:temperature">
                <swe:uom code="Cel"/>
                <swe:value>22.3</swe:value>
              </swe:Quantity>
            </om:parameter>
            <om:result xsi:type="swe:QuantityPropertyType">
              <swe:Quantity definition=
                "urn:ogc:def:property:OGC:1.0:mass">
                <swe:uom code="kg"/>
                <swe:value>0.28</swe:value>
              </swe:Quantity>
            </om:result>
          </om:Observation>
        </wsnt:Message>
      </wsnt:NotificationMessage>
    </wsnt:Notify>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.5.2 RegisterPublisher operation

Via this operation, new publishers – usually sensors – can be added to the SES.

10.5.2.1 Request

The operation request shall be implemented and used as defined in the WS-BrokeredNotification operation. However, a SensorML descriptionshall be added to the request using the extension mechanisms provided in the syntax of the operation request. In addition, the topic shall be a child of Measurements as defined by the SES TopicSet (see section 8.5).

The WS-Addressing (action) Message Addressing Property shall – as specified in WS-BrokeredNotification - contain the URI:
http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest

The body of an exemplary request is shown in the following listing.

Listing 12: RegisterPublisher request example

```
<?xml version="1.0" encoding="UTF-8"?>
<wsn-br:RegisterPublisher
xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <wsn-br:Topic Dialect=
     "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
       ses:Measurements
  </wsn-br:Topic>
  <sml:SensorML version="1.0.1">
    <sml:member xlink:role="urn:ogc:def:role:OGC:sensorSystem">
      <sml:System>
        <!-- system contents here -->
      </sml:System>
    </sml:member>
  </sml:SensorML>
</wsn-br:RegisterPublisher>
```

The SES shall be responsible for publishing messages on the sensor management and expiration topics, not the registered sensor, thus a RegisterPublisher request that is sent to an SES and which contains a SensorML description shall provide the ses:Measurements topic in its topic expression.

10.5.2.2 Response

The operation response shall be implemented and used as defined in the WS-BrokeredNotification operation.

The WS-Addressing (action) Message Addressing Property shall – as specified in WS-BrokeredNotification - contain the URI:
http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherResponse

The body of an exemplary response is shown in the following listing.

Listing 13: RegisterPublisher operation response example

```
<?xml version="1.0" encoding="UTF-8"?>
<wsn-br:RegisterPublisherResponse
xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <wsn-br:PublisherRegistrationReference>
    <wsa:Address>
http://my.ses.org/SES/services/SesPublisherRegistrationManager
      </wsa:Address>
    <wsa:ReferenceParameters>
<!-- any specific properties for the given reference, e.g. an ID
-->
    </wsa:ReferenceParameters>
  </wsn-br:PublisherRegistrationReference>
</wsn-br:RegisterPublisherResponse>
```

10.6 SesPublisherRegistrationManager Port Type

This port type combines the RenewRegistration and DestroyRegistration operations.

10.6.1 RenewRegistration operation

The RenewRegistration operation shall be implemented and used as defined in the Renew operation from the WS-BaseNotification specification for renewing subscriptions, the only difference being that the purpose of the request shall renew a registration (of a publisher) and not a subscription.

10.6.1.1 Request

The RenewRegistration request shall be sent to the endpoint provided in the RegisterPublisher operation response in the PublisherRegistrationReference element. The requestor suggests a new termination time by either providing a xs:dateTime (interpreted as absolute termination time) or by providing a duration (which shall be added to the current time of the service receiving the request to form an absolute time).

The WS-Addressing (action) Message Addressing Property shall contain the URI: http://www.opengis.net/ses/SesPublisherRegistrationManager/RenewRegistrationRequest

The structure of the operation request is shown in Figure 13.

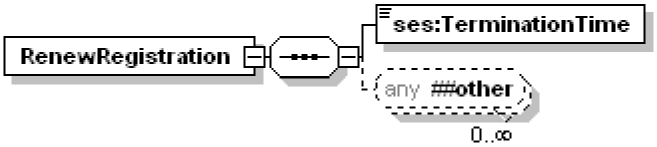


Figure 13: RenewRegistration operation request structure

10.6.1.2 Response

If the request is accepted by the service, a RenewRegistrationResponse shall be returned which provides the new termination time (given as dateTime). In addition, the service may also provide its current time (as xs:dateTime) so that the client knows when the response was generated (and thus when the termination time was reset to the given value).

The WS-Addressing (action) Message Addressing Property shall contain the URI: http://www.opengis.net/ses/SesPublisherRegistrationManager/RenewRegistrationResponse

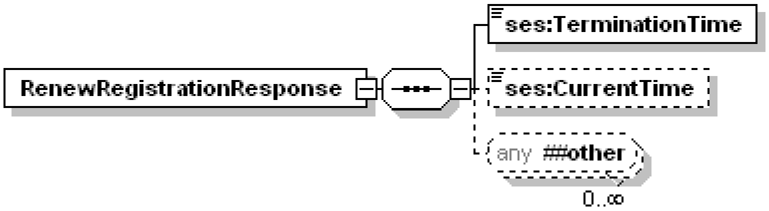


Figure 14: RenewRegistration operation response structure

10.6.2 DestroyRegistration operation

The DestroyRegistration operation shall be implemented and used as defined in the WS-BrokeredNotification specification.

10.7 WS-ResourceProperties operations

The operations defined by WS-ResourceProperties are: GetResourceProperty, GetResourcePropertyDocument, GetMultipleResourceProperties, QueryResourceProperties, PutResourcePropertyDocument, SetResourceProperties, InsertResourceProperties, UpdateResourceProperties and DeleteResourceProperties.

All these operations shall be implemented and used as defined in the WS-ResourceProperties specification.

10.8 WS-ResourceLifetime Operations

The operations defined by WS-ResourceLifetime are: Destroy and SetTerminationTime.

All these operations shall be implemented and used as defined in the WS-ResourceLifetime specification.

Annex A
(normative)

Abstract test suite

An abstract test suite is not provided in this version of this Implementation Specification, but will be provided in version 1.0.0.

Annex B
(normative)

XML Schema Documents

In addition to this document, this specification includes several normative XML Schema Documents.

B.1 - sesAll.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
  <xs:annotation>
    <xs:appinfo>sesAll.xsd</xs:appinfo>
    <xs:documentation>
      <description>This XML Schema includes, directly and
indirectly, all the XML Schemas defined by the Sensor Event
Service (SES).</description>
    </xs:documentation>
  </xs:annotation>
  <!--
==============================================================
      includes and imports
   ==============================================================
-->
  <xs:include schemaLocation="sesCapabilities.xsd"/>
  <xs:include schemaLocation="sesDescribeSensor.xsd"/>
  <xs:include schemaLocation="sesRenewRegistration.xsd"/>
  <xs:include schemaLocation="sesMessageSchema.xsd"/>
</xs:schema>
```

B.2 – sesCapabilities.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:ogc="http://www.opengis.net/ogc"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
```

```
  <!--
==============================================================
      includes and imports
   ==============================================================
-->
  <xs:import namespace="http://www.opengis.net/ows/1.1"
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/
>
  <xs:import namespace="http://www.opengis.net/ogc"
schemaLocation="../filter/1.1.0_experimental/filterCapabilities.x
sd"/>
  <xs:include schemaLocation="sesCommon.xsd"/>
  <!--
==============================================================
      elements and types
   ==============================================================
-->
  <xs:element name="GetCapabilities">
    <xs:annotation>
      <xs:documentation>Request to a SES to perform the
GetCapabilities operation. This operation allows a client to
retrieve service metadata (capabilities XML) providing metadata
for the specific SES server. In this XML encoding, no "request"
parameter is included, since the element name specifies the
specific operation. </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ows:GetCapabilitiesType">
          <xs:attribute name="service" type="ows:ServiceType"
use="required" fixed="SES"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <!--
=========================================================== -->
  <xs:element name="Capabilities">
    <xs:annotation>
      <xs:documentation>XML encoded SES GetCapabilities operation
response. This document provides clients with service metadata
about a specific service instance, including metadata about the
tightly-coupled data served. If the server does not implement the
updateSequence parameter, the server shall always return the
complete Capabilities document, without the updateSequence
parameter. When the server implements the updateSequence
parameter and the GetCapabilities operation request included the
updateSequence parameter with the current value, the server shall
return this element with only the "version" and "updateSequence"
attributes. Otherwise, all optional elements shall be included or
not depending on the actual value of the Sections parameter in
the GetCapabilities operation request. </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ows:CapabilitiesBaseType">
          <xs:sequence>
            <xs:element ref="ses:Contents" minOccurs="0"/>
            <xs:element ref="ogc:Filter_Capabilities"
minOccurs="0"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Contents">
    <xs:annotation>
      <xs:documentation/>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="registeredSensors" minOccurs="0">
          <xs:annotation>
            <xs:documentation>root element for all
SubscriptionOfferings</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ses:SensorID"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B.3 – sesCommon.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
  <xs:annotation>
    <xs:appinfo>sesCommon.xsd</xs:appinfo>
    <xs:documentation>
         This XML Schema encodes the elements and types that are
shared by muliple SES operations.
      </xs:documentation>
  </xs:annotation>
  <!--
=============================================================
      elements and types
   =============================================================
-->
  <xs:complexType name="RequestBaseType">
    <xs:annotation>
      <xs:documentation>XML encoded SES operation request base.
In this XML encoding, no "request" parameter is included, since
the element name specifies the specific operation.
</xs:documentation>
    </xs:annotation>
    <xs:attribute name="service" type="xs:string" use="required"
fixed="SES">
      <xs:annotation>
        <xs:documentation>Service type identifier.
</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="version" type="xs:string" use="required"
fixed="1.0.0">
      <xs:annotation>
        <xs:documentation>Specification version for SES version
and operation.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:element name="SensorID" type="xs:token">
    <xs:annotation>
      <xs:documentation>Unique ID for every registered
sensor.</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:schema>
```

B.4 – sesDescribeSensor.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ses="http://www.opengis.net/ses/0.0"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
  <!--
=============================================================
      includes and imports
   =============================================================
-->
  <xs:include schemaLocation="sesCommon.xsd"/>
  <!--
=============================================================
      request
   ==============================================================
-->
  <xs:element name="DescribeSensor">
    <xs:annotation>
      <xs:documentation>Used to request detailed sensor
metadata.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ses:RequestBaseType">
          <xs:sequence>
            <xs:element ref="ses:SensorID"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <!--The response is a SensorML document.-->
</xs:schema>
```

B.5 – sesEvent.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml"
xmlns:eml="http://www.opengis.net/eml/0.0"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
  <xs:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd
"/>
  <xs:import namespace="http://www.opengis.net/eml/0.0"
schemaLocation="../08-132/eml.xsd"/>
  <xs:annotation>
    <xs:appinfo>sesEvent.xsd</xs:appinfo>
    <xs:documentation>
         This XML Schema encodes the element and type of a
SESEvent.
      </xs:documentation>
  </xs:annotation>
  <!--
==============================================================
      elements and types
   ==============================================================
-->
  <xs:element name="SESEvent" type="ses:SESEventType"
substitutionGroup="gml:_GML">
```

```
    <xs:annotation>
      <xs:documentation>Representation of an event, used for
complex event processing and event stream
processing</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="SESEventType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
        <xs:sequence>
          <xs:element ref="eml:Event"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

B.6 – sesMessageSchema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="SensorManagementInformation">
    <xs:annotation>
      <xs:documentation>Contains information about a sensor which
reached a new publisher state.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SensorID" type="xs:anyURI">
          <xs:annotation>
            <xs:documentation>ID of the sensor which is in a new
status.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="SensorStatus" type="xs:token">
          <xs:annotation>
            <xs:documentation>Contains information about the new
status of a sensor. Refer to the specification to learn about
allowed values and their meanings (registered, paused, resumed,
unregistered,...)</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="description" type="xs:string"
minOccurs="0">
          <xs:annotation>
```

```
            <xs:documentation>May contain additional information
about the status change, for example that a sensor was
unregistered due to the fact that its registration
expired.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SensorExpirationInformation">
    <xs:annotation>
      <xs:documentation>Provides information about when a
specific sensors registration will expire. How often this kind of
message is generated depends on the service implementation (but
should be indicated through policies).</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SensorID" type="xs:anyURI"/>
        <xs:element name="ServerTime" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Current time at the service when it
generated the message.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ExpirationTime" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Point in time at which the sensor
registration will expire. Note that the registration may be
renewed after this type of message was created. Therefore this
information reflects the state of the service at the given
ServerTime.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B.7 – sesRenewRegistration.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ses="http://www.opengis.net/ses/0.0"
targetNamespace="http://www.opengis.net/ses/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xml:lang="en">
  <!--
==============================================================
      request
```

```
   ==============================================================
-->
  <xs:element name="RenewRegistration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="TerminationTime"
type="ses:AbsoluteOrRelativeTimeType" nillable="true"/>
        <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--
==============================================================
      response
   ==============================================================
-->
  <xs:element name="RenewRegistrationResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="TerminationTime" type="xs:dateTime"/>
        <xs:element name="CurrentTime" type="xs:dateTime"
minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="AbsoluteOrRelativeTimeType">
    <xs:union memberTypes="xs:dateTime xs:duration"/>
  </xs:simpleType>
</xs:schema>
```

Annex C
(informative)

WSDL examples

Based upon the interfaces and operations defined in (and referenced by) this specification, WSDL descriptions of service interfaces can be constructed.

C.1 – WSDL for Sensor Event Service

The example shows an interface description implementing operations from WS-ResourceProperties, WS-ResourceLifetime, WS-BaseNotification, WS-BrokeredNotification and SES.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://www.opengis.net/ses/0.0"
    xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl-soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsrf-
r="http://docs.oasis-open.org/wsrf/r-2"
    xmlns:wsrf-rl="http://docs.oasis-open.org/wsrf/rl-2"
    xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
    xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/rp-2"
    xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
    xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
    xmlns:wsn-t="http://docs.oasis-open.org/wsn/t-1"
    xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
    name="SES">

    <wsdl:types>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.w3.org/1999/xlink">
            <xsd:include
schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd
"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="urn:us:gov:ic:ism:v2">
            <xsd:include
schemaLocation="http://schemas.opengis.net/ic/2.0/IC-ISM-
v2.xsd"/>
```

```
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"

targetNamespace="http://www.w3.org/2001/SMIL20/Language">
            <xsd:include

schemaLocation="http://schemas.opengis.net/gml/3.1.1/smil/smil20-
language.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.w3.org/2001/SMIL20/">
            <xsd:include
schemaLocation="http://schemas.opengis.net/gml/3.1.1/smil/smil20.
xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ows/1.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/
>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/gml">
            <xsd:include
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd
"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ogc">
            <xsd:include
schemaLocation="../xsd/filter/1.1.0_experimental/filter.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/eml/0.0">
            <xsd:include schemaLocation="../xsd/08-132/eml.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://www.opengis.net/swe/1.0.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/sweCommon/1.0.1/swe.xs
d"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"

targetNamespace="http://www.opengis.net/sensorML/1.0.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/sensorML/1.0.1/sensorM
L.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"

targetNamespace="http://www.w3.org/2005/08/addressing">
```

```
            <xsd:include
schemaLocation="http://www.w3.org/2005/08/addressing/ws-
addr.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/rl-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/rl-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/rp-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/rp-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/r-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/r-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/b-2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/b-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/t-1">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/t-1.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/br-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/br-2.xsd"/>
        </xsd:schema>

        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ses/0.0">
            <xsd:include schemaLocation="../xsd/ses/sesAll.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesCapabilities.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesCommon.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesDescribeSensor.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesRenewRegistration.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesMessageSchema.xsd"/>
```

```
            <xsd:element name="SesResourceProperties">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="wsn-b:FixedTopicSet"/>
                        <xsd:element ref="wsn-t:TopicSet"
minOccurs="0"/>
                        <xsd:element ref="wsn-b:TopicExpression"
minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element ref="wsn-
b:TopicExpressionDialect" minOccurs="0"
                            maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>

            <xsd:complexType name="ServiceExceptionType">
                <xsd:complexContent>
                    <xsd:extension base="wsrf-bf:BaseFaultType">
                        <xsd:choice>
                            <xsd:element
ref="ows:ExceptionReport"/>
                            <xsd:element ref="ows:Exception"/>
                        </xsd:choice>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>

            <xsd:element name="ServiceException"
type="ses:ServiceExceptionType"/>

        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="DestroyRequest">
        <wsdl:part name="DestroyRequest" element="wsrf-
rl:Destroy"/>
    </wsdl:message>
    <wsdl:message name="DestroyResponse">
        <wsdl:part name="DestroyResponse" element="wsrf-
rl:DestroyResponse"/>
    </wsdl:message>
    <wsdl:message name="ResourceNotDestroyedFault">
        <wsdl:part name="ResourceNotDestroyedFault"
element="wsrf-rl:ResourceNotDestroyedFault"/>
    </wsdl:message>
    <wsdl:message name="ResourceUnknownFault">
        <wsdl:part name="ResourceUnknownFault" element="wsrf-
r:ResourceUnknownFault"/>
    </wsdl:message>
    <wsdl:message name="ResourceUnavailableFault">
        <wsdl:part name="ResourceUnavailableFault" element="wsrf-
r:ResourceUnavailableFault"/>
```

```
    </wsdl:message>
    <wsdl:message name="SetTerminationTimeRequest">
        <wsdl:part name="SetTerminationTimeRequest"
element="wsrf-rl:SetTerminationTime"/>
    </wsdl:message>
    <wsdl:message name="SetTerminationTimeResponse">
        <wsdl:part name="SetTerminationTimeResponse"
element="wsrf-rl:SetTerminationTimeResponse"/>
    </wsdl:message>
    <wsdl:message name="UnableToSetTerminationTimeFault">
        <wsdl:part name="UnableToSetTerminationTimeFault"
            element="wsrf-rl:UnableToSetTerminationTimeFault"/>
    </wsdl:message>
    <wsdl:message name="TerminationTimeChangeRejectedFault">
        <wsdl:part name="TerminationTimeChangeRejectedFault"
            element="wsrf-
rl:TerminationTimeChangeRejectedFault"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyDocumentRequest">
        <wsdl:part name="GetResourcePropertyDocumentRequest"
            element="wsrf-rp:GetResourcePropertyDocument"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyDocumentResponse">
        <wsdl:part name="GetResourcePropertyDocumentResponse"
            element="wsrf-
rp:GetResourcePropertyDocumentResponse"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyRequest">
        <wsdl:part name="GetResourcePropertyRequest"
element="wsrf-rp:GetResourceProperty"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyResponse">
        <wsdl:part name="GetResourcePropertyResponse"
element="wsrf-rp:GetResourcePropertyResponse"
        />
    </wsdl:message>
    <wsdl:message name="InvalidResourcePropertyQNameFault">
        <wsdl:part name="InvalidResourcePropertyQNameFault"
            element="wsrf-rp:InvalidResourcePropertyQNameFault"/>
    </wsdl:message>
    <wsdl:message name="GetMultipleResourcePropertiesRequest">
        <wsdl:part name="GetMultipleResourcePropertiesRequest"
            element="wsrf-rp:GetMultipleResourceProperties"/>
    </wsdl:message>
    <wsdl:message name="GetMultipleResourcePropertiesResponse">
        <wsdl:part name="GetMultipleResourcePropertiesResponse"
            element="wsrf-
rp:GetMultipleResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="QueryResourcePropertiesRequest">
        <wsdl:part name="QueryResourcePropertiesRequest"
element="wsrf-rp:QueryResourceProperties"/>
```

```
    </wsdl:message>
    <wsdl:message name="QueryResourcePropertiesResponse">
        <wsdl:part name="QueryResourcePropertiesResponse"
            element="wsrf-rp:QueryResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="UnknownQueryExpressionDialectFault">
        <wsdl:part name="UnknownQueryExpressionDialectFault"
            element="wsrf-
rp:UnknownQueryExpressionDialectFault"/>
    </wsdl:message>
    <wsdl:message name="InvalidQueryExpressionFault">
        <wsdl:part name="InvalidQueryExpressionFault"
element="wsrf-rp:InvalidQueryExpressionFault"
        />
    </wsdl:message>
    <wsdl:message name="QueryEvaluationErrorFault">
        <wsdl:part name="QueryEvaluationErrorFault"
element="wsrf-rp:QueryEvaluationErrorFault"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertiesRequest">
        <wsdl:part name="SetResourcePropertiesRequest"
element="wsrf-rp:SetResourceProperties"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertiesResponse">
        <wsdl:part name="SetResourcePropertiesResponse"
            element="wsrf-rp:SetResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="InvalidModificationFault">
        <wsdl:part name="InvalidModificationFault" element="wsrf-
rp:InvalidModificationFault"/>
    </wsdl:message>
    <wsdl:message name="UnableToModifyResourcePropertyFault">
        <wsdl:part name="UnableToModifyResourcePropertyFault"
            element="wsrf-
rp:UnableToModifyResourcePropertyFault"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertyRequestFailedFault">
        <wsdl:part name="SetResourcePropertyRequestFailedFault"
            element="wsrf-
rp:SetResourcePropertyRequestFailedFault"/>
    </wsdl:message>
    <wsdl:message name="SubscribeRequest">
        <wsdl:part name="SubscribeRequest" element="wsn-
b:Subscribe"/>
    </wsdl:message>

    <wsdl:message name="SubscribeResponse">
        <wsdl:part name="SubscribeResponse" element="wsn-
b:SubscribeResponse"/>
    </wsdl:message>

    <wsdl:message name="SubscribeCreationFailedFault">
```

```
        <wsdl:part name="SubscribeCreationFailedFault"
element="wsn-b:SubscribeCreationFailedFault"
        />
    </wsdl:message>

    <wsdl:message name="TopicExpressionDialectUnknownFault">
        <wsdl:part name="TopicExpressionDialectUnknownFault"
            element="wsn-b:TopicExpressionDialectUnknownFault"/>
    </wsdl:message>

    <wsdl:message name="InvalidFilterFault">
        <wsdl:part name="InvalidFilterFault" element="wsn-
b:InvalidFilterFault"/>
    </wsdl:message>

    <wsdl:message
name="InvalidProducerPropertiesExpressionFault">
        <wsdl:part
name="InvalidProducerPropertiesExpressionFault"
            element="wsn-
b:InvalidProducerPropertiesExpressionFault"/>
    </wsdl:message>

    <wsdl:message name="InvalidMessageContentExpressionFault">
        <wsdl:part name="InvalidMessageContentExpressionFault"
            element="wsn-
b:InvalidMessageContentExpressionFault"/>
    </wsdl:message>

    <wsdl:message name="UnrecognizedPolicyRequestFault">
        <wsdl:part name="UnrecognizedPolicyRequestFault"
            element="wsn-b:UnrecognizedPolicyRequestFault"/>
    </wsdl:message>

    <wsdl:message name="UnsupportedPolicyRequestFault">
        <wsdl:part name="UnsupportedPolicyRequestFault"
            element="wsn-b:UnsupportedPolicyRequestFault"/>
    </wsdl:message>

    <wsdl:message name="NotifyMessageNotSupportedFault">
        <wsdl:part name="NotifyMessageNotSupportedFault"
            element="wsn-b:NotifyMessageNotSupportedFault"/>
    </wsdl:message>

    <wsdl:message name="UnacceptableInitialTerminationTimeFault">
        <wsdl:part name="UnacceptableInitialTerminationTimeFault"
            element="wsn-
b:UnacceptableInitialTerminationTimeFault"/>
    </wsdl:message>

    <wsdl:message name="GetCurrentMessageRequest">
```

```
        <wsdl:part name="GetCurrentMessageRequest" element="wsn-
b:GetCurrentMessage"/>
    </wsdl:message>

    <wsdl:message name="GetCurrentMessageResponse">
        <wsdl:part name="GetCurrentMessageResponse" element="wsn-
b:GetCurrentMessageResponse"/>
    </wsdl:message>

    <wsdl:message name="InvalidTopicExpressionFault">
        <wsdl:part name="InvalidTopicExpressionFault"
element="wsn-b:InvalidTopicExpressionFault"/>
    </wsdl:message>

    <wsdl:message name="TopicNotSupportedFault">
        <wsdl:part name="TopicNotSupportedFault" element="wsn-
b:TopicNotSupportedFault"/>
    </wsdl:message>

    <wsdl:message name="MultipleTopicsSpecifiedFault">
        <wsdl:part name="MultipleTopicsSpecifiedFault"
element="wsn-b:MultipleTopicsSpecifiedFault"
        />
    </wsdl:message>

    <wsdl:message name="NoCurrentMessageOnTopicFault">
        <wsdl:part name="NoCurrentMessageOnTopicFault"
element="wsn-b:NoCurrentMessageOnTopicFault"
        />
    </wsdl:message>

    <wsdl:message name="Notify">
        <wsdl:part name="Notify" element="wsn-b:Notify"/>
    </wsdl:message>
    <wsdl:message name="RegisterPublisherRequest">
        <wsdl:part name="RegisterPublisherRequest" element="wsn-
br:RegisterPublisher"/>
    </wsdl:message>

    <wsdl:message name="RegisterPublisherResponse">
        <wsdl:part name="RegisterPublisherResponse" element="wsn-
br:RegisterPublisherResponse"/>
    </wsdl:message>


    <!-- ============= MESSAGES ============= -->

    <!-- SES specific messages -->
    <wsdl:message name="GetCapabilitiesRequest">
        <wsdl:part name="GetCapabilitiesRequest"
element="ses:GetCapabilities"/>
    </wsdl:message>
    <wsdl:message name="GetCapabilitiesResponse">
```

```
        <wsdl:part name="GetCapabilitiesResponse"
element="ses:Capabilities"/>
    </wsdl:message>

    <wsdl:message name="DescribeSensorRequest">
        <wsdl:part name="DescribeSensorRequest"
element="ses:DescribeSensor"/>
    </wsdl:message>
    <wsdl:message name="DescribeSensorResponse">
        <wsdl:part name="DescribeSensorResponse"
element="sml:SensorML"/>
    </wsdl:message>

    <wsdl:message name="ServiceException">
        <wsdl:part name="ServiceException"
type="ses:OWSServiceException"/>
    </wsdl:message>

    <!-- =================================================== -->
    <!-- ==================== SES PortType ==================== -
->
    <!-- =================================================== -->
    <wsdl:portType name="SesPortType" wsrf-
rp:ResourceProperties="ses:SesResourceProperties">

        <!-- ============= implements GetCapabilities
============= -->
        <wsdl:operation name="GetCapabilities">
            <wsdl:input
wsa:Action="http://www.opengis.net/ses/GetCapabilitiesRequest"
                message="ses:GetCapabilitiesRequest"/>
            <wsdl:output
wsa:Action="http://www.opengis.net/ses/GetCapabilitiesResponse"
                message="ses:GetCapabilitiesResponse"/>
            <wsdl:fault name="Exception"
message="ses:ServiceException"/>
        </wsdl:operation>

        <!-- ============= implements DescribeSensor
============= -->
        <wsdl:operation name="DescribeSensor">
            <wsdl:input
wsa:Action="http://www.opengis.net/ses/DescribeSensorRequest"
                message="ses:DescribeSensorRequest"/>
            <wsdl:output
wsa:Action="http://www.opengis.net/ses/DescribeSensorResponse"
                message="ses:DescribeSensorResponse"/>
            <wsdl:fault name="Exception"
message="ses:ServiceException"/>
        </wsdl:operation>
        <!-- ============= extends NotificationConsumer
============= -->
        <wsdl:operation name="Notify">
            <wsdl:input wsa:Action="http://docs.oasis-
open.org/wsn/bw-2/NotificationConsumer/Notify"
                message="ses:Notify"/>
        </wsdl:operation>

        <!-- ============= extends NotificationProducer
============= -->
        <wsdl:operation name="Subscribe">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeRequest"
                message="ses:SubscribeRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeResponse"
                message="ses:SubscribeResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="InvalidFilterFault"
message="ses:InvalidFilterFault"/>
            <wsdl:fault name="TopicExpressionDialectUnknownFault"

message="ses:TopicExpressionDialectUnknownFault"/>
            <wsdl:fault name="InvalidTopicExpressionFault"
message="ses:InvalidTopicExpressionFault"/>
            <wsdl:fault name="TopicNotSupportedFault"
message="ses:TopicNotSupportedFault"/>
            <wsdl:fault
name="InvalidProducerPropertiesExpressionFault"

message="ses:InvalidProducerPropertiesExpressionFault"/>
            <wsdl:fault
name="InvalidMessageContentExpressionFault"

message="ses:InvalidMessageContentExpressionFault"/>
            <wsdl:fault
name="UnacceptableInitialTerminationTimeFault"

message="ses:UnacceptableInitialTerminationTimeFault"/>
            <wsdl:fault name="UnrecognizedPolicyRequestFault"
                message="ses:UnrecognizedPolicyRequestFault"/>
            <wsdl:fault name="UnsupportedPolicyRequestFault"
                message="ses:UnsupportedPolicyRequestFault"/>
            <wsdl:fault name="NotifyMessageNotSupportedFault"
                message="ses:NotifyMessageNotSupportedFault"/>
            <wsdl:fault name="SubscribeCreationFailedFault"
                message="ses:SubscribeCreationFailedFault"/>
        </wsdl:operation>

        <wsdl:operation name="GetCurrentMessage">
```

```
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/GetCurrentMessageRequest"
                message="ses:GetCurrentMessageRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/GetCurrentMessageResponse"
                message="ses:GetCurrentMessageResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="TopicExpressionDialectUnknownFault"

message="ses:TopicExpressionDialectUnknownFault"/>
            <wsdl:fault name="InvalidTopicExpressionFault"
message="ses:InvalidTopicExpressionFault"/>
            <wsdl:fault name="TopicNotSupportedFault"
message="ses:TopicNotSupportedFault"/>
            <wsdl:fault name="NoCurrentMessageOnTopicFault"
                message="ses:NoCurrentMessageOnTopicFault"/>
            <wsdl:fault name="MultipleTopicsSpecifiedFault"
                message="ses:MultipleTopicsSpecifiedFault"/>
        </wsdl:operation>

        <!-- ========= extends RegisterPublisher ======= -->
        <!-- Will be used by SES to both register push-style
sensors but also to register pull-style for SOS
         (via extension mechanism - has to be specified) -->
        <wsdl:operation name="RegisterPublisher">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsn/brw-
2/RegisterPublisher/RegisterPublisherRequest"
                message="ses:RegisterPublisherRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsn/brw-
2/RegisterPublisher/RegisterPublisherResponse"
                message="ses:RegisterPublisherResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="InvalidTopicExpressionFault"
message="ses:InvalidTopicExpressionFault"/>
            <wsdl:fault name="TopicNotSupportedFault"
message="ses:TopicNotSupportedFault"/>
            <wsdl:fault name="PublisherRegistrationRejectedFault"

message="ses:PublisherRegistrationRejectedFault"/>
            <wsdl:fault name="PublisherRegistrationFailedFault"
                message="ses:PublisherRegistrationFailedFault"/>
            <wsdl:fault
name="UnacceptableInitialTerminationTimeFault"

message="ses:UnacceptableInitialTerminationTimeFault"/>
        </wsdl:operation>
        <!-- =========== implements WS-ResourceProperties
operations =========== -->
        <wsdl:operation name="GetResourcePropertyDocument">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourcePropertyDocument/GetResourcePropertyDocumentRequest"
                name="GetResourcePropertyDocumentRequest"

message="ses:GetResourcePropertyDocumentRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourcePropertyDocument/GetResourcePropertyDocumentResponse
"
                name="GetResourcePropertyDocumentResponse"

message="ses:GetResourcePropertyDocumentResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
        </wsdl:operation>

        <wsdl:operation name="GetResourceProperty">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourceProperty/GetResourcePropertyRequest"
                name="GetResourcePropertyRequest"
message="ses:GetResourcePropertyRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourceProperty/GetResourcePropertyResponse"
                name="GetResourcePropertyResponse"
message="ses:GetResourcePropertyResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
                message="ses:InvalidResourcePropertyQNameFault"/>
        </wsdl:operation>

        <wsdl:operation name="GetMultipleResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetMultipleResourceProperties/GetMultipleResourcePropertiesRequ
est"
                name="GetMultipleResourcePropertiesRequest"

message="ses:GetMultipleResourcePropertiesRequest"/>
            <wsdl:output
```

```
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetMultipleResourceProperties/GetMultipleResourcePropertiesResp
onse"
                name="GetMultipleResourcePropertiesResponse"

message="ses:GetMultipleResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
                message="ses:InvalidResourcePropertyQNameFault"/>
        </wsdl:operation>

        <wsdl:operation name="QueryResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/QueryResourceProperties/QueryResourcePropertiesRequest"
                name="QueryResourcePropertiesRequest"
message="ses:QueryResourcePropertiesRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/QueryResourceProperties/QueryResourcePropertiesResponse"
                name="QueryResourcePropertiesResponse"
message="ses:QueryResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="UnknownQueryExpressionDialectFault"

message="ses:UnknownQueryExpressionDialectFault"/>
            <wsdl:fault name="InvalidQueryExpressionFault"
message="ses:InvalidQueryExpressionFault"/>
            <wsdl:fault name="QueryEvaluationErrorFault"
message="ses:QueryEvaluationErrorFault"/>
        </wsdl:operation>

        <wsdl:operation name="SetResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/SetResourceProperties/SetResourcePropertiesRequest"
                name="SetResourcePropertiesRequest"
message="ses:SetResourcePropertiesRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/SetResourceProperties/SetResourcePropertiesResponse"
                name="SetResourcePropertiesResponse"
message="ses:SetResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
```

```
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidModificationFault"
message="ses:InvalidModificationFault"/>
            <wsdl:fault
name="UnableToModifyResourcePropertyFault"

message="ses:UnableToModifyResourcePropertyFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
                message="ses:InvalidResourcePropertyQNameFault"/>
            <wsdl:fault
name="SetResourcePropertyRequestFailedFault"

message="ses:SetResourcePropertyRequestFailedFault"/>
        </wsdl:operation>
    </wsdl:portType>

    <!-- ============================= =========================
-->
    <!-- ====================== BINDING =========================
-->
    <!-- =======================================================
-->
    <wsdl:binding name="SesBinding" type="ses:SesPortType">
        <wsdl-soap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

        <wsdl:operation name="GetCapabilities">
            <wsdl-soap12:operation soapAction="GetCapabilities"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="Exception">
                <wsdl-soap12:fault use="encoded"
name="Exception"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="DescribeSensor">
            <wsdl-soap12:operation soapAction="DescribeSensor"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output>
```

```
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="Exception">
                <wsdl-soap12:fault use="encoded"
name="Exception"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="Notify">
            <wsdl-soap12:operation soapAction="Notify"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
        </wsdl:operation>

        <wsdl:operation name="Subscribe">
            <wsdl-soap12:operation soapAction="Subscribe"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidFilterFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidFilterFault"/>
            </wsdl:fault>
            <wsdl:fault
name="TopicExpressionDialectUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="TopicExpressionDialectUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidTopicExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidTopicExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault name="TopicNotSupportedFault">
                <wsdl-soap12:fault use="encoded"
name="TopicNotSupportedFault"/>
            </wsdl:fault>
            <wsdl:fault
name="InvalidProducerPropertiesExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidProducerPropertiesExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault
name="InvalidMessageContentExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidMessageContentExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnacceptableInitialTerminationTimeFault">
                <wsdl-soap12:fault use="encoded"
name="UnacceptableInitialTerminationTimeFault"/>
            </wsdl:fault>
            <wsdl:fault name="UnrecognizedPolicyRequestFault">
                <wsdl-soap12:fault use="encoded"
name="UnrecognizedPolicyRequestFault"/>
            </wsdl:fault>
            <wsdl:fault name="UnsupportedPolicyRequestFault">
                <wsdl-soap12:fault use="encoded"
name="UnsupportedPolicyRequestFault"/>
            </wsdl:fault>
            <wsdl:fault name="NotifyMessageNotSupportedFault">
                <wsdl-soap12:fault use="encoded"
name="NotifyMessageNotSupportedFault"/>
            </wsdl:fault>
            <wsdl:fault name="SubscribeCreationFailedFault">
                <wsdl-soap12:fault use="encoded"
name="SubscribeCreationFailedFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetCurrentMessage">
            <wsdl-soap12:operation
soapAction="GetCurrentMessage"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault
name="TopicExpressionDialectUnknownFault">
```

```
                <wsdl-soap12:fault use="encoded"
name="TopicExpressionDialectUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidTopicExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidTopicExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault name="TopicNotSupportedFault">
                <wsdl-soap12:fault use="encoded"
name="TopicNotSupportedFault"/>
            </wsdl:fault>
            <wsdl:fault name="NoCurrentMessageOnTopicFault">
                <wsdl-soap12:fault use="encoded"
name="NoCurrentMessageOnTopicFault"/>
            </wsdl:fault>
            <wsdl:fault name="MultipleTopicsSpecifiedFault">
                <wsdl-soap12:fault use="encoded"
name="MultipleTopicsSpecifiedFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="RegisterPublisher">
            <wsdl-soap12:operation
soapAction="RegisterPublisher"/>
            <wsdl:input>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output>
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidTopicExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidTopicExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault name="TopicNotSupportedFault">
                <wsdl-soap12:fault use="encoded"
name="TopicNotSupportedFault"/>
            </wsdl:fault>
            <wsdl:fault
name="PublisherRegistrationRejectedFault">
                <wsdl-soap12:fault use="encoded"
name="PublisherRegistrationRejectedFault"/>
            </wsdl:fault>
            <wsdl:fault name="PublisherRegistrationFailedFault">
```

```
                <wsdl-soap12:fault use="encoded"
name="PublisherRegistrationFailedFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnacceptableInitialTerminationTimeFault">
                <wsdl-soap12:fault use="encoded"
name="UnacceptableInitialTerminationTimeFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetResourcePropertyDocument">
            <wsdl-soap12:operation
soapAction="GetResourcePropertyDocument"/>
            <wsdl:input
name="GetResourcePropertyDocumentRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output
name="GetResourcePropertyDocumentResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetResourceProperty">
            <wsdl-soap12:operation
soapAction="GetResourceProperty"/>
            <wsdl:input name="GetResourcePropertyRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="GetResourcePropertyResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
```

```
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetMultipleResourceProperties">
            <wsdl-soap12:operation
soapAction="GetMultipleResourceProperties"/>
            <wsdl:input
name="GetMultipleResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output
name="GetMultipleResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="QueryResourceProperties">
            <wsdl-soap12:operation
soapAction="QueryResourceProperties"/>
            <wsdl:input name="QueryResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="QueryResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
```

```
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnknownQueryExpressionDialectFault">
                <wsdl-soap12:fault use="encoded"
name="UnknownQueryExpressionDialectFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidQueryExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidQueryExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault name="QueryEvaluationErrorFault">
                <wsdl-soap12:fault use="encoded"
name="QueryEvaluationErrorFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="SetResourceProperties">
            <wsdl-soap12:operation
soapAction="http://oasis.org/SetResourceProperties"/>
            <wsdl:input name="SetResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="SetResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidModificationFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidModificationFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnableToModifyResourcePropertyFault">
                <wsdl-soap12:fault use="encoded"
name="UnableToModifyResourcePropertyFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
```

```
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
            </wsdl:fault>
            <wsdl:fault
name="SetResourcePropertyRequestFailedFault">
                <wsdl-soap12:fault use="encoded"
name="SetResourcePropertyRequestFailedFault"/>
            </wsdl:fault>
        </wsdl:operation>

    </wsdl:binding>
    <wsdl:service name="SesService">
        <wsdl:port name="SesPort" binding="ses:SesBinding">
            <wsdl-soap12:address
location="http://localhost:8080/SES/services/ses"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

C.2 – WSDL for SES Publisher Registration Manager

The PRM is an interface containing operations to manage publisher registrations. See clause 10.6 for further details.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://www.opengis.net/ses/0.0"
    xmlns:ses="http://www.opengis.net/ses/0.0"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl-soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsrf-
r="http://docs.oasis-open.org/wsrf/r-2"
    xmlns:wsrf-rl="http://docs.oasis-open.org/wsrf/rl-2"
    xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
    xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/rp-2"
    xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
    xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
    xmlns:wsn-t="http://docs.oasis-open.org/wsn/t-1"
    xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
    name="SES">

    <wsdl:types>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.w3.org/1999/xlink">
```

```
            <xsd:include
schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd
"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="urn:us:gov:ic:ism:v2">
            <xsd:include
schemaLocation="http://schemas.opengis.net/ic/2.0/IC-ISM-
v2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"

targetNamespace="http://www.w3.org/2001/SMIL20/Language">
            <xsd:include

schemaLocation="http://schemas.opengis.net/gml/3.1.1/smil/smil20-
language.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.w3.org/2001/SMIL20/">
            <xsd:include
schemaLocation="http://schemas.opengis.net/gml/3.1.1/smil/smil20.
xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ows/1.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/
>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/gml">
            <xsd:include
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd
"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ogc">
            <xsd:include
schemaLocation="../xsd/filter/1.1.0_experimental/filter.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/eml/0.0">
            <xsd:include schemaLocation="../xsd/08-132/eml.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://www.opengis.net/swe/1.0.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/sweCommon/1.0.1/swe.xs
d"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
```

```
targetNamespace="http://www.opengis.net/sensorML/1.0.1">
            <xsd:include
schemaLocation="http://schemas.opengis.net/sensorML/1.0.1/sensorM
L.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"

targetNamespace="http://www.w3.org/2005/08/addressing">
            <xsd:include
schemaLocation="http://www.w3.org/2005/08/addressing/ws-
addr.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/rl-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/rl-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/rp-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/rp-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsrf/r-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsrf/r-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/b-2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/b-2.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/t-1">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/t-1.xsd"/>
        </xsd:schema>
        <xsd:schema elementFormDefault="qualified"
            targetNamespace="http://docs.oasis-open.org/wsn/br-
2">
            <xsd:include schemaLocation="http://docs.oasis-
open.org/wsn/br-2.xsd"/>
        </xsd:schema>

        <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/ses/0.0">
```

```
            <xsd:include
schemaLocation="../xsd/ses/sesCommon.xsd"/>
            <xsd:include
schemaLocation="../xsd/ses/sesRenewRegistration.xsd"/>

            <xsd:element name="SesPRMResourceProperties">
                <xsd:complexType>
                    <xsd:sequence>
                        <!-- The resource contains the properties
defined for a PublisherRegistration resource-->
                        <xsd:element ref="wsrf-rl:CurrentTime"/>
                        <xsd:element ref="wsrf-
rl:TerminationTime"/>

                        <!-- The resource contains the properties
defined for a PublisherRegistration resource -->
                        <xsd:element ref="wsn-
br:PublisherReference" minOccurs="0" maxOccurs="1"/>
                        <xsd:element ref="wsn-br:Topic"
minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element ref="wsn-br:Demand"
minOccurs="1" maxOccurs="1"/>
                        <xsd:element ref="wsn-br:CreationTime"
minOccurs="0" maxOccurs="1"/>

                        <!-- The resource also provides the
SensorML description of the registered publisher if it is a
sensor. -->
                        <xsd:element ref="sml:SensorML"
minOccurs="0" maxOccurs="1"/>

                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="DestroyRegistrationRequest">
        <wsdl:part name="DestroyRegistrationRequest"
element="wsn-br:DestroyRegistration"/>
    </wsdl:message>
    <wsdl:message name="DestroyRegistrationResponse">
        <wsdl:part name="DestroyRegistrationResponse"
element="wsn-br:DestroyRegistrationResponse"/>
    </wsdl:message>
    <wsdl:message name="DestroyRequest">
        <wsdl:part name="DestroyRequest" element="wsrf-
rl:Destroy"/>
    </wsdl:message>
    <wsdl:message name="DestroyResponse">
        <wsdl:part name="DestroyResponse" element="wsrf-
rl:DestroyResponse"/>
    </wsdl:message>
    <wsdl:message name="ResourceNotDestroyedFault">
        <wsdl:part name="ResourceNotDestroyedFault"
element="wsrf-rl:ResourceNotDestroyedFault"/>
    </wsdl:message>
    <wsdl:message name="WSNBRResourceNotDestroyedFault">
        <wsdl:part name="WSNBRResourceNotDestroyedFault"
element="wsn-br:ResourceNotDestroyedFault"
        />
    </wsdl:message>
    <wsdl:message name="ResourceUnknownFault">
        <wsdl:part name="ResourceUnknownFault" element="wsrf-
r:ResourceUnknownFault"/>
    </wsdl:message>
    <wsdl:message name="ResourceUnavailableFault">
        <wsdl:part name="ResourceUnavailableFault" element="wsrf-
r:ResourceUnavailableFault"/>
    </wsdl:message>
    <wsdl:message name="SetTerminationTimeRequest">
        <wsdl:part name="SetTerminationTimeRequest"
element="wsrf-rl:SetTerminationTime"/>
    </wsdl:message>
    <wsdl:message name="SetTerminationTimeResponse">
        <wsdl:part name="SetTerminationTimeResponse"
element="wsrf-rl:SetTerminationTimeResponse"/>
    </wsdl:message>
    <wsdl:message name="UnableToSetTerminationTimeFault">
        <wsdl:part name="UnableToSetTerminationTimeFault"
            element="wsrf-rl:UnableToSetTerminationTimeFault"/>
    </wsdl:message>
    <wsdl:message name="TerminationTimeChangeRejectedFault">
        <wsdl:part name="TerminationTimeChangeRejectedFault"
            element="wsrf-
rl:TerminationTimeChangeRejectedFault"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyDocumentRequest">
        <wsdl:part name="GetResourcePropertyDocumentRequest"
            element="wsrf-rp:GetResourcePropertyDocument"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyDocumentResponse">
        <wsdl:part name="GetResourcePropertyDocumentResponse"
            element="wsrf-
rp:GetResourcePropertyDocumentResponse"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyRequest">
        <wsdl:part name="GetResourcePropertyRequest"
element="wsrf-rp:GetResourceProperty"/>
    </wsdl:message>
    <wsdl:message name="GetResourcePropertyResponse">
        <wsdl:part name="GetResourcePropertyResponse"
element="wsrf-rp:GetResourcePropertyResponse"
        />
    </wsdl:message>
    <wsdl:message name="InvalidResourcePropertyQNameFault">
        <wsdl:part name="InvalidResourcePropertyQNameFault"
            element="wsrf-rp:InvalidResourcePropertyQNameFault"/>
    </wsdl:message>
    <wsdl:message name="GetMultipleResourcePropertiesRequest">
        <wsdl:part name="GetMultipleResourcePropertiesRequest"
            element="wsrf-rp:GetMultipleResourceProperties"/>
    </wsdl:message>
    <wsdl:message name="GetMultipleResourcePropertiesResponse">
        <wsdl:part name="GetMultipleResourcePropertiesResponse"
            element="wsrf-
rp:GetMultipleResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="QueryResourcePropertiesRequest">
        <wsdl:part name="QueryResourcePropertiesRequest"
element="wsrf-rp:QueryResourceProperties"/>
    </wsdl:message>
    <wsdl:message name="QueryResourcePropertiesResponse">
        <wsdl:part name="QueryResourcePropertiesResponse"
            element="wsrf-rp:QueryResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="UnknownQueryExpressionDialectFault">
        <wsdl:part name="UnknownQueryExpressionDialectFault"
            element="wsrf-
rp:UnknownQueryExpressionDialectFault"/>
    </wsdl:message>
    <wsdl:message name="InvalidQueryExpressionFault">
        <wsdl:part name="InvalidQueryExpressionFault"
element="wsrf-rp:InvalidQueryExpressionFault"
        />
    </wsdl:message>
    <wsdl:message name="QueryEvaluationErrorFault">
        <wsdl:part name="QueryEvaluationErrorFault"
element="wsrf-rp:QueryEvaluationErrorFault"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertiesRequest">
        <wsdl:part name="SetResourcePropertiesRequest"
element="wsrf-rp:SetResourceProperties"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertiesResponse">
        <wsdl:part name="SetResourcePropertiesResponse"
            element="wsrf-rp:SetResourcePropertiesResponse"/>
    </wsdl:message>
    <wsdl:message name="InvalidModificationFault">
        <wsdl:part name="InvalidModificationFault" element="wsrf-
rp:InvalidModificationFault"/>
    </wsdl:message>
    <wsdl:message name="UnableToModifyResourcePropertyFault">
        <wsdl:part name="UnableToModifyResourcePropertyFault"
            element="wsrf-
rp:UnableToModifyResourcePropertyFault"/>
    </wsdl:message>
    <wsdl:message name="SetResourcePropertyRequestFailedFault">
```

```
        <wsdl:part name="SetResourcePropertyRequestFailedFault"
            element="wsrf-
rp:SetResourcePropertyRequestFailedFault"/>
    </wsdl:message>

    <wsdl:message name="UnacceptableInitialTerminationTimeFault">
        <wsdl:part name="UnacceptableInitialTerminationTimeFault"
            element="wsn-
b:UnacceptableInitialTerminationTimeFault"/>
    </wsdl:message>

    <wsdl:message name="RenewRegistrationRequest">
        <wsdl:part name="RenewRegistrationRequest"
element="ses:RenewRegistration"/>
    </wsdl:message>
    <wsdl:message name="RenewRegistrationResponse">
        <wsdl:part name="RenewRegistrationResponse"
element="ses:RenewRegistrationResponse"/>
    </wsdl:message>

    <!-- =================================================== -->
    <!-- ======== SES PublisherRegistrationManager
PortType======== -->
    <!-- =================================================== -->
    <wsdl:portType name="SesPRMPortType" wsrf-
rp:ResourceProperties="ses:SesPRMResourceProperties">

        <!-- =========== implements WS-
PublisherRegistrationManager interface =========== -->
        <wsdl:operation name="DestroyRegistration">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsn/brw-
2/PublisherRegistrationManager/DestroyRegistrationRequest"
                name="DestroyRegistrationRequest"
message="ses:DestroyRegistrationRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsn/brw-
2/PublisherRegistrationManager/DestroyRegistrationResponse"
                name="DestroyRegistrationResponse"
message="ses:DestroyRegistrationResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="WSNBRResourceNotDestroyedFault"
                message="ses:WSNBRResourceNotDestroyedFault"/>
        </wsdl:operation>

        <!-- =========== implements the following WS-
ResourceProperties operations =========== -->
        <wsdl:operation name="GetResourcePropertyDocument">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourcePropertyDocument/GetResourcePropertyDocumentRequest"
```

```
                name="GetResourcePropertyDocumentRequest"

message="ses:GetResourcePropertyDocumentRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourcePropertyDocument/GetResourcePropertyDocumentResponse
"
                name="GetResourcePropertyDocumentResponse"

message="ses:GetResourcePropertyDocumentResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
        </wsdl:operation>

        <wsdl:operation name="GetResourceProperty">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourceProperty/GetResourcePropertyRequest"
                name="GetResourcePropertyRequest"
message="ses:GetResourcePropertyRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetResourceProperty/GetResourcePropertyResponse"
                name="GetResourcePropertyResponse"
message="ses:GetResourcePropertyResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
                message="ses:InvalidResourcePropertyQNameFault"/>
        </wsdl:operation>

        <wsdl:operation name="GetMultipleResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetMultipleResourceProperties/GetMultipleResourcePropertiesRequ
est"
                name="GetMultipleResourcePropertiesRequest"

message="ses:GetMultipleResourcePropertiesRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/GetMultipleResourceProperties/GetMultipleResourcePropertiesResp
onse"
                name="GetMultipleResourcePropertiesResponse"

message="ses:GetMultipleResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
```

```
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
                message="ses:InvalidResourcePropertyQNameFault"/>
        </wsdl:operation>

        <wsdl:operation name="QueryResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/QueryResourceProperties/QueryResourcePropertiesRequest"
                name="QueryResourcePropertiesRequest"
message="ses:QueryResourcePropertiesRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/QueryResourceProperties/QueryResourcePropertiesResponse"
                name="QueryResourcePropertiesResponse"
message="ses:QueryResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="UnknownQueryExpressionDialectFault"

message="ses:UnknownQueryExpressionDialectFault"/>
            <wsdl:fault name="InvalidQueryExpressionFault"
message="ses:InvalidQueryExpressionFault"/>
            <wsdl:fault name="QueryEvaluationErrorFault"
message="ses:QueryEvaluationErrorFault"/>
        </wsdl:operation>

        <wsdl:operation name="SetResourceProperties">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/SetResourceProperties/SetResourcePropertiesRequest"
                name="SetResourcePropertiesRequest"
message="ses:SetResourcePropertiesRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rpw-
2/SetResourceProperties/SetResourcePropertiesResponse"
                name="SetResourcePropertiesResponse"
message="ses:SetResourcePropertiesResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="InvalidModificationFault"
message="ses:InvalidModificationFault"/>
            <wsdl:fault
name="UnableToModifyResourcePropertyFault"

message="ses:UnableToModifyResourcePropertyFault"/>
            <wsdl:fault name="InvalidResourcePropertyQNameFault"
```

```
                message="ses:InvalidResourcePropertyQNameFault"/>
            <wsdl:fault
name="SetResourcePropertyRequestFailedFault"

message="ses:SetResourcePropertyRequestFailedFault"/>
        </wsdl:operation>

        <!-- implements the WS-ResourceLifetime operations -->
        <wsdl:operation name="Destroy">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rlw-
2/ImmediateResourceTermination/DestroyRequest"
                name="DestroyRequest"
message="ses:DestroyRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rlw-
2/ImmediateResourceTermination/DestroyResponse"
                name="DestroyResponse"
message="ses:DestroyResponse"/>
            <wsdl:fault name="ResourceNotDestroyedFault"
message="ses:ResourceNotDestroyedFault"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
        </wsdl:operation>
        <wsdl:operation name="SetTerminationTime">
            <wsdl:input
                wsa:Action="http://docs.oasis-open.org/wsrf/rlw-
2/ScheduledResourceTermination/SetTerminationTimeRequest"
                name="SetTerminationTimeRequest"
message="ses:SetTerminationTimeRequest"/>
            <wsdl:output
                wsa:Action="http://docs.oasis-open.org/wsrf/rlw-
2/ScheduledResourceTermination/SetTerminationTimeResponse"
                name="SetTerminationTimeResponse"
message="ses:SetTerminationTimeResponse"/>
            <wsdl:fault name="UnableToSetTerminationTimeFault"
                message="ses:UnableToSetTerminationTimeFault"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="ResourceUnavailableFault"
message="ses:ResourceUnavailableFault"/>
            <wsdl:fault name="TerminationTimeChangeRejectedFault"

message="ses:TerminationTimeChangeRejectedFault"/>
        </wsdl:operation>

        <!-- ========= implements the Renew operation ======= -->
        <wsdl:operation name="RenewRegistration">
            <wsdl:input
```

```
wsa:Action="http://www.opengis.net/ses/SesPublisherRegistrationMa
nager/RenewRegistrationRequest"
                name="RenewRegistrationRequest"
message="ses:RenewRegistrationRequest"/>
            <wsdl:output

wsa:Action="http://www.opengis.net/ses/SesPublisherRegistrationMa
nager/RenewRegistrationResponse"
                name="RenewRegistrationResponse"
message="ses:RenewRegistrationResponse"/>
            <wsdl:fault name="ResourceUnknownFault"
message="ses:ResourceUnknownFault"/>
            <wsdl:fault name="UnacceptableTerminationTimeFault"
                message="ses:UnacceptableTerminationTimeFault"/>
        </wsdl:operation>

    </wsdl:portType>

    <!-- ============================= =========================
-->
    <!-- ====================== BINDING =========================
-->
    <!-- =======================================================
-->
    <wsdl:binding name="SesPRMBinding" type="ses:SesPRMPortType">
        <wsdl-soap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

        <wsdl:operation name="RenewRegistration">
            <wsdl-soap12:operation
soapAction="RenewRegistration"/>
            <wsdl:input name="RenewRegistrationRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="RenewRegistrationResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="UnacceptableTerminationTimeFault">
                <wsdl-soap12:fault use="encoded"
name="UnacceptableTerminationTimeFault"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="DestroyRegistration">
            <wsdl-soap12:operation
soapAction="DestroyRegistration"/>
            <wsdl:input name="DestroyRegistrationRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="DestroyRegistrationResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="WSNBRResourceNotDestroyedFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceNotDestroyedFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="Destroy">
            <wsdl-soap12:operation soapAction="Destroy"/>
            <wsdl:input name="DestroyRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="DestroyResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceNotDestroyedFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceNotDestroyedFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="SetTerminationTime">
            <wsdl-soap12:operation
soapAction="SetTerminationTime"/>
            <wsdl:input name="SetTerminationTimeRequest">
                <wsdl-soap12:body use="literal"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="SetTerminationTimeResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="UnableToSetTerminationTimeFault">
                <wsdl-soap12:fault use="encoded"
name="UnableToSetTerminationTimeFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault
name="TerminationTimeChangeRejectedFault">
                <wsdl-soap12:fault use="encoded"
name="TerminationTimeChangeRejectedFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetResourcePropertyDocument">
            <wsdl-soap12:operation
soapAction="GetResourcePropertyDocument"/>
            <wsdl:input
name="GetResourcePropertyDocumentRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output
name="GetResourcePropertyDocumentResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="GetResourceProperty">
            <wsdl-soap12:operation
soapAction="GetResourceProperty"/>
            <wsdl:input name="GetResourcePropertyRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="GetResourcePropertyResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="GetMultipleResourceProperties">
            <wsdl-soap12:operation
soapAction="GetMultipleResourceProperties"/>
            <wsdl:input
name="GetMultipleResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output
name="GetMultipleResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
```

```
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="QueryResourceProperties">
            <wsdl-soap12:operation
soapAction="QueryResourceProperties"/>
            <wsdl:input name="QueryResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="QueryResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnknownQueryExpressionDialectFault">
                <wsdl-soap12:fault use="encoded"
name="UnknownQueryExpressionDialectFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidQueryExpressionFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidQueryExpressionFault"/>
            </wsdl:fault>
            <wsdl:fault name="QueryEvaluationErrorFault">
                <wsdl-soap12:fault use="encoded"
name="QueryEvaluationErrorFault"/>
            </wsdl:fault>
        </wsdl:operation>

        <wsdl:operation name="SetResourceProperties">
            <wsdl-soap12:operation
soapAction="http://oasis.org/SetResourceProperties"/>
            <wsdl:input name="SetResourcePropertiesRequest">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:input>
            <wsdl:output name="SetResourcePropertiesResponse">
                <wsdl-soap12:body use="literal"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </wsdl:output>
            <wsdl:fault name="ResourceUnknownFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnknownFault"/>
            </wsdl:fault>
            <wsdl:fault name="ResourceUnavailableFault">
                <wsdl-soap12:fault use="encoded"
name="ResourceUnavailableFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidModificationFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidModificationFault"/>
            </wsdl:fault>
            <wsdl:fault
name="UnableToModifyResourcePropertyFault">
                <wsdl-soap12:fault use="encoded"
name="UnableToModifyResourcePropertyFault"/>
            </wsdl:fault>
            <wsdl:fault name="InvalidResourcePropertyQNameFault">
                <wsdl-soap12:fault use="encoded"
name="InvalidResourcePropertyQNameFault"/>
            </wsdl:fault>
            <wsdl:fault
name="SetResourcePropertyRequestFailedFault">
                <wsdl-soap12:fault use="encoded"
name="SetResourcePropertyRequestFailedFault"/>
            </wsdl:fault>
        </wsdl:operation>

    </wsdl:binding>
    <wsdl:service name="SesPRMService">
        <wsdl:port name="SesPRMPort" binding="ses:SesPRMBinding">
            <wsdl-soap12:address

location="http://localhost:8080/SES/services/SesPublisherRegistra
tionManager"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Bibliography

[1] OpenGIS® Implementation Specification *Sensor Observation Service. OGC Document 06-009r6*