Open Geospatial Consortium Inc.

Date: 2005-10-10

Reference number of this OGC® document: **OGC 06-028**

Version: 0.2.0

Category: OpenGIS® Discussion Paper

Editor: Ingo Simonis

OpenGIS® Sensor Alert Service Implementation Specification

Warning

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

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

Document type:	OpenGIS® Implementation Specification
Document subtype:	if applicable
Document stage:	Public Discussion Paper
Document language:	English

Contents

Page

Figures

Page

Tables

Page

i. Preface

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. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification

iii. Submitting organizations

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

University of Muenster

Oak Ridge National Laboratory

3eti

iv. Document contributor contact points

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

Name	Organization
Mark Priest	3eti
Ingo Simonis (editor)	University of Muenster
Johnny Tolliver	Oak Ridge National Laboratory
Alexander Walkowski	University of Muenster

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2005-12-28	0.1.0	Ingo Simonis		Initial version
2006-01-03	0.1.1	Alexander Walkowski	All	Examples added and general revision
2006-02-12	0.2.0	Ingo Simonis	All	Final changes to submit this version in time of the three week rule for the OGC TC meeting March 2006.

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document[IS1][jst2].

vii. Future work

This work depicts the initial version of the Sensor Alert Service. Future modifications are likely to appear in the near future.

Foreword

The Sensor Alert Service is the newest part of the OGC Sensor Web Enablement document suite.

This document includes four annexes; Annexes A and B are normative, and Annexes C and D are informative.

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

Introduction

The Sensor Alert Service (SAS) can be compared with an event notification system. The sensor node is the object of interest. Each node has to advertise its publications at a SAS (*advertise*). Note: Actually, it is the "alerts" that a sensor can send that must be registered rather than the node/sensor itself. It is important to notice that both, sensors that just send the current observation value as an alert and those sensors that are able to send alerts like "above a HIGH threshold" and "below a LOW threshold" can register at the SAS. This allows more “clever” sensors to be attached to SAS as well as those that only send current observations. There is no difference made between those types of sensors! If the sensor just sends its current observation value, it is up to the SAS to check its subscription table if the current value is above or below a user defined threshold. The same applies to those alerts that are sent from sensors that can decide on their own if a threshold is exceeded/felt short of.

All alert types get registered, or advertised, that they can potentially be sent. If an event occurs the node will send it to the SAS via the *publish* operation. A consumer (interested party) may *subscribe* to alerts disseminated by the SAS. If an event occurs the SAS will *notify* all clients subscribed to this event type.

Figure 1: Event Notification System

Traditional OGC web services are not suitable for implementing this alert service. Although the XMPP protocol is widely used within this specification (and we used it intensively during our initial tests), we emphasize that the SAS specification is agnostic regarding the used alert protocol.

In order to emphasize the deployment of SAS servers, we highly recommend to use XMPP as the messaging protocol!

OpenGIS® Sensor Alert Service Implementation Specification

1 Scope

This OpenGIS® document specifies interfaces for requesting information describing the capabilities of a Sensor Alert Service, for determining the nature of offered alerts, the protocols used, and the options to subscribe to specific alert types.

2 Conformance

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

3 Normative references

The following normative documents contain provisions 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.

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

OGC 05-008, *OpenGIS® Web Services Common Specification*

This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

OGC 05-114 *OpenGIS®* Web Notification Service

OGC 05-090 SWE Architecture

OGC 05-088 SOS

OGC 05-087 O&M

OGC 05-086 SensorML

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 05-008] shall apply. In addition, the following terms and definitions apply.

4.1 alert[jst3]

An alert is a special kind of notification indicating that an event has occurred at an object of interest, which results in a condition of heightened watchfulness or preparation for action. Alerts do always contain a time and location value.

4.2 notification

A message sent to a receiver indicating some kind of event.

5 Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

SOS	Sensor Observation Service
WNS	Web Notification Service
SAS	Sensor Alert Service
SWE	Sensor Web Enablement
O&M	Observation and Measurement
SensorML	Sensor Model Language
TML	Transducer Markup Language

5.2 UML notation

Most diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 05-008].

5.3 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 accompanied UML and schema perfectly.

---

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

5.3.1 Element

A named rectangle represents 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.

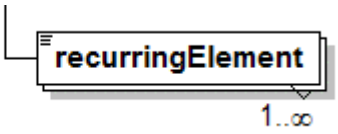
This is a mandatory simple element. Note the upper left corner of the rectangle indicates that data is contained in this element.

5.3.2 Optional Element

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

5.3.3 Recurring Element

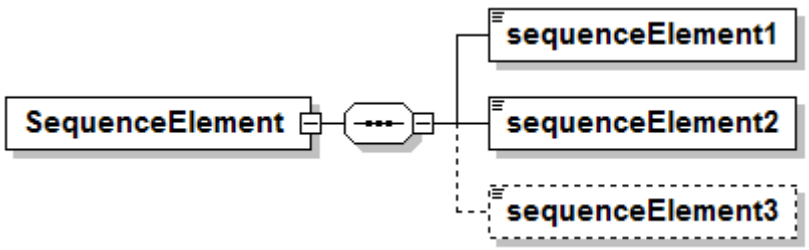
This element (and its child elements if it has any) can occur multiple times.



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.

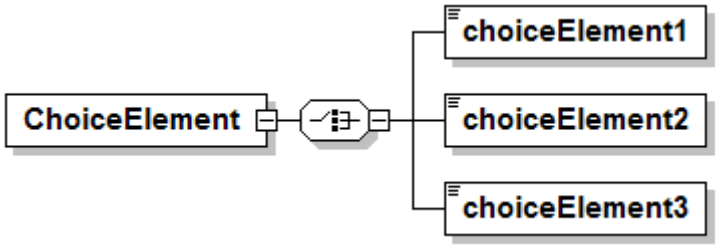
5.3.4 Sequence Connector

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



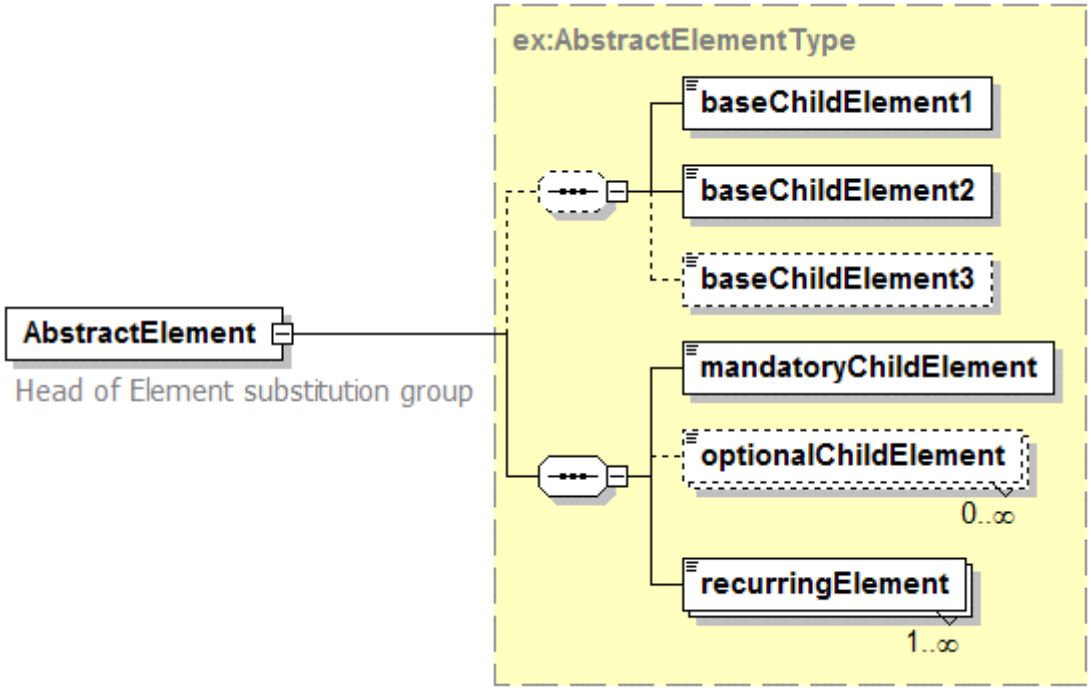
5.3.5 Choice Connector

The connection box here is a “choice” indicator, indicating that there is always going to be exactly one of the child elements listed on the right.



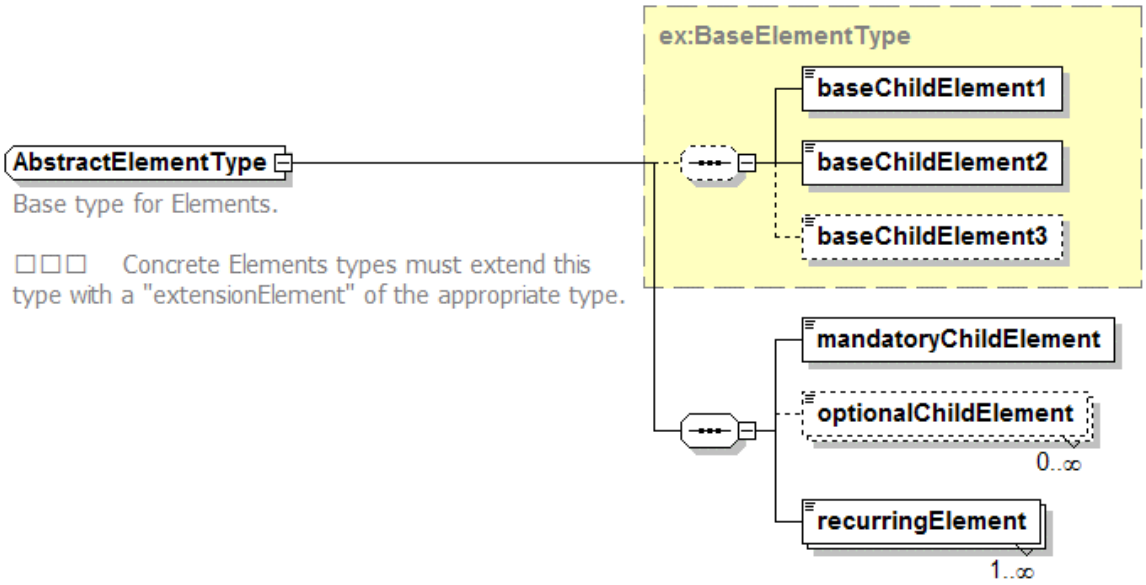
5.3.6 Definition with Complex Type

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



5.3.7 Complex Type

This diagram illustrates the definition of a complex type (i.e., “AbstractElementType”), extending another complex type (i.e.,“ex:BaseElementType”) with three additional elements. Complex types can be reused to specify that different elements are of the same type.



5.4 Used parts of other documents

This document uses significant parts of document [OGC 05-008]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are accompanied by a NOTE.

5.5 Platform-neutral and platform-specific specifications

As specified in Clause 10 of OGC Abstract Specification Topic 12 “OpenGIS Service Architecture” (which contains ISO 19119), this document includes both Distributed Computing Platform-neutral and platform-specific specifications. This document first specifies each operation request and response in platform-neutral fashion. This is done using a table for each data structure, which lists and defines the parameters and other data structures contained. These tables serve as data dictionaries for the UML model in Annex C, and thus specify the UML model data type and multiplicity of each listed item.

The specified platform-neutral data could be encoded in many alternative ways, each appropriate to one or more specific distributed computing platforms (DCPs). This document now specifies encoding appropriate for use of HTTP GET transfer of operations requests (using key-value-pair (KVP) encoding), and for use of HTTP POST transfer of operations requests (using XML encoding).

6 SAS overview

The specified SAS specifies an interface that allows nodes to advertise and publish observational data or its describing metadata respectively. It is important to emphasize that the SAS itself acts rather like a registry than an event notification system! Sensors or other data producers do advertise their offers to a messaging server. The messaging server itself forwards this advertisement to the SAS. If a consumer wants to subscribe to an alert, it sends a subscription-request to the SAS. We want to point out that this operation is rather a lookup than a real subscription. This is based on the fact that the SAS will not send any alerts. All messaging is performed by a messaging server (keep in mind that we are talking about logical instances at this time. This means that Messaging Server and SAS may work on the same machine, both of them may even work in the very same memory space). The response send by the SAS will contain the communication endpoint. It is up to the consumer to open a connection to this communication endpoint. The SAS response contains all information necessary to set up a subscription. The following three figures illustrate SAS and messaging server activity. The first diagram illustrates a sensor advertising its publications.

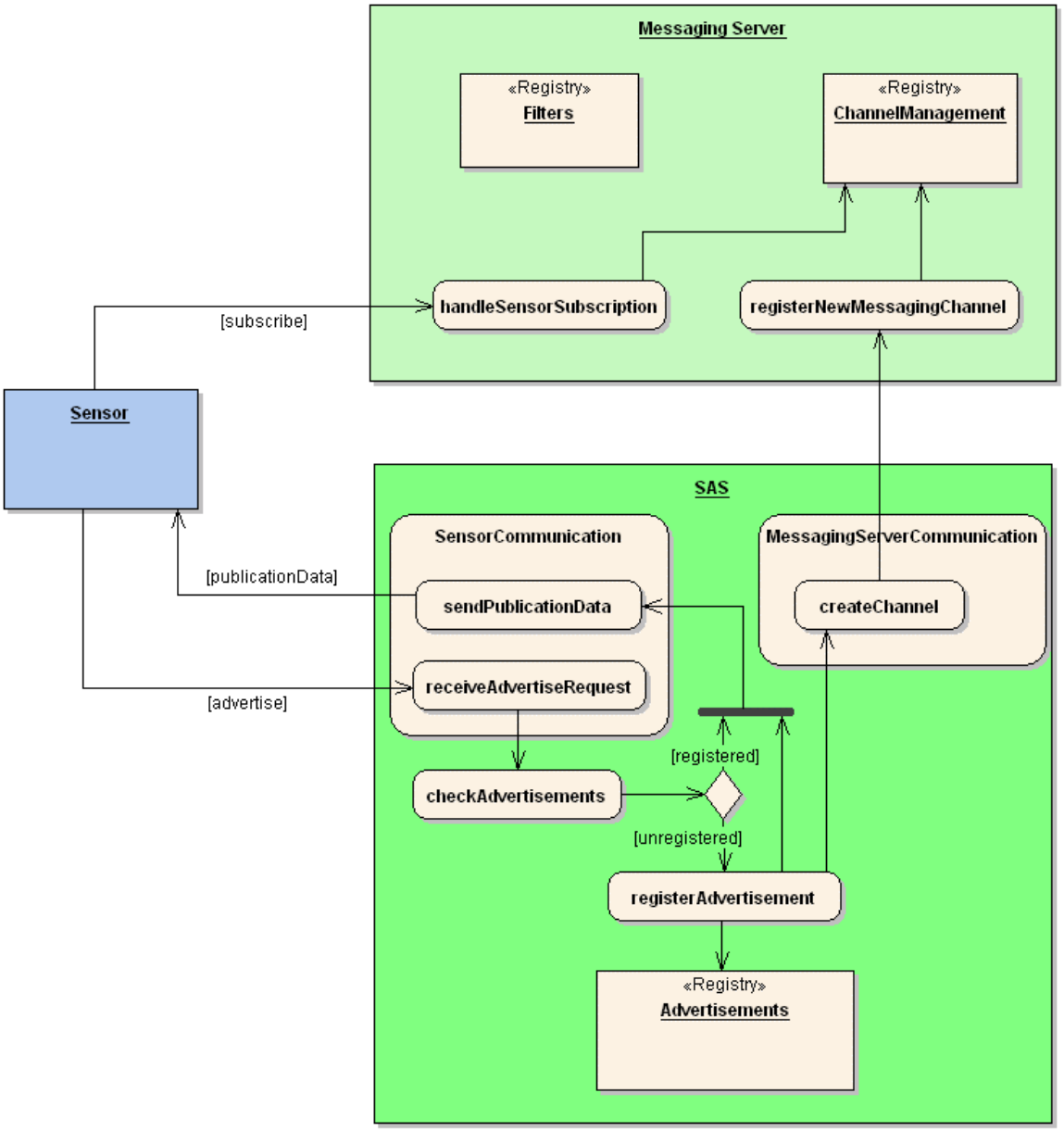


Figure 2: Overview sensor publication and consumer subscription

The next figure illustrates a consumer subscribing to an alert. The consumer first contacts the SAS in order to retrieve the necessary information for subscribing at the messaging server.

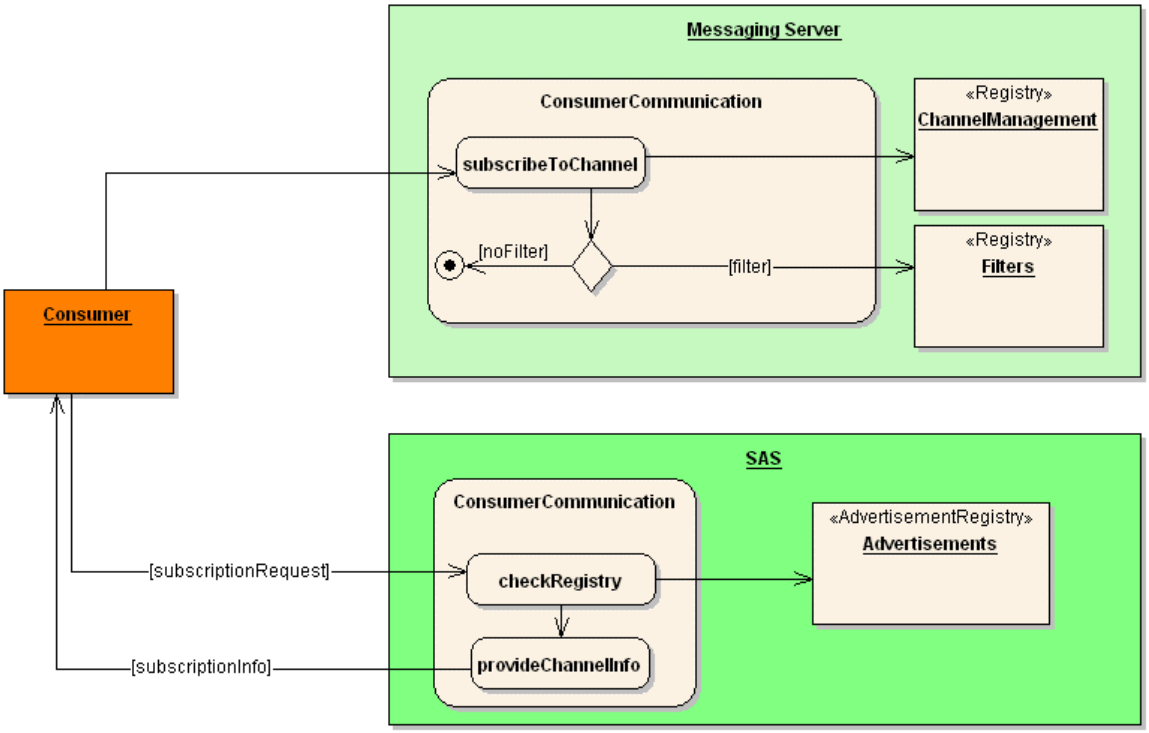


Figure 3: Overview consumer subscription

The next figure illustrates a sensor publishing an observation. Note that this publication has to be in accordance to the previous sensor advertisement. This means that if the sensor has advertised that it can only publish “temperature data”, it cannot send a “battery low” alert, or rather this publication will be discarded by the messaging system. Note that the SAS is NOT involved in the sensor publication / user notification process. The publication provided by the sensor will be checked by the messaging server if a consumer has subscribed to it. Additionally, it will be analysed if the consumer has set an additional filter. If both conditions are true, a notification will be sent to the consumer.

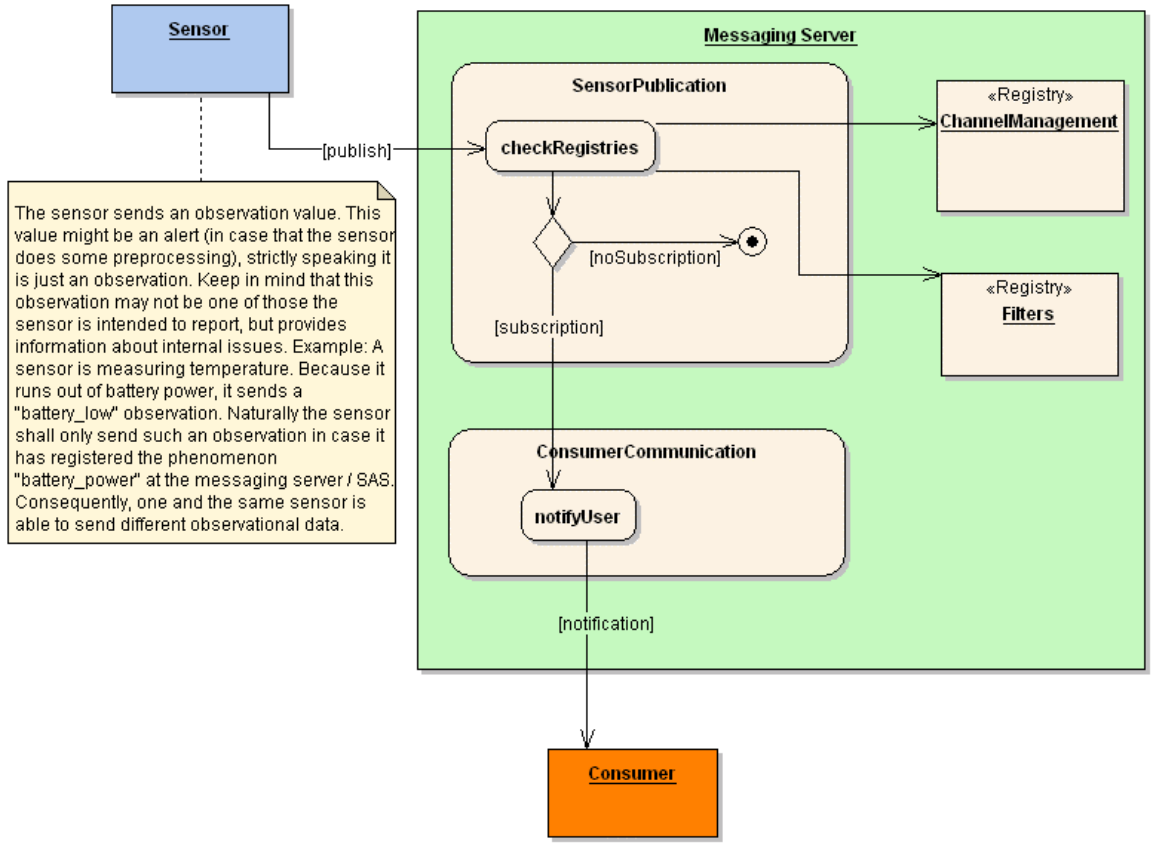


Figure 4: Overview sensor publication

The SAS interface (currently) specifies seven operations that can be requested by a client and performed by a SAS server. Those operations are:

a) GetCapabilities (required implementation by servers) – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. This operation also supports negotiation of the specification version being used for client-server interactions.

b) Advertise (required implementation by servers) – This operation allows producers to advertise the type of information published. In case that this data is the product of some pre-processing, e.g. sensor announces a *battery low* status, this data might be considered as an alert. In fact, it is only a published observation!

c) CancelAdvertisement (required implementation by servers) – This operation allows producers to cancel an advertisement.

d) RenewAdvertisement (required implementation by servers) – This operation allows producers to renew an advertisement

e) Subscribe (required implementation by servers) – This operation allows consumers to subscribe to alerts (keep in mind that this depicts a virtual subscription; the real subscription takes place at the messaging service interface).

f) CancelSubscription (required implementation by servers) – This operation allows consumers to cancel a subscription[IS4].

g) RenewSubscription (required implementation by servers) – This operation allows consumers to renew a subscription[IS5].

Figure 1 is a simple UML diagram summarizing the SAS interface. This class diagram shows that the SAS interface class inherits the getCapabilities operation from the OGCWebService interface class, and adds the SAS operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the SAS interface is provided in Annex C (informative).

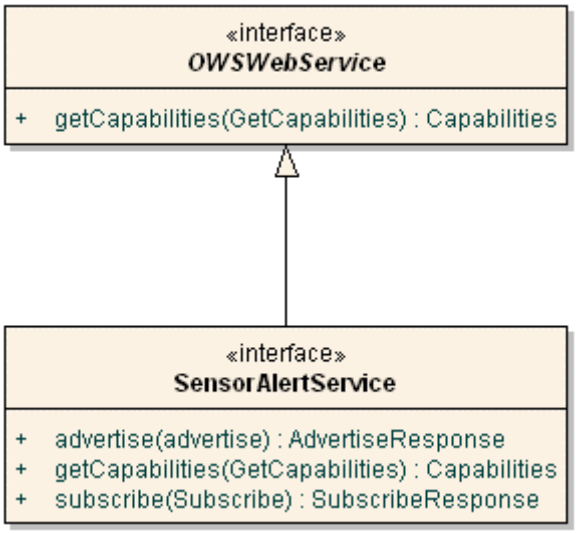


Figure 5: SAS interface UML diagram

NOTE In this UML diagram, the request and response for each operation is shown as a single parameter that is a data structure containing multiple lower-level parameters, which are discussed in subsequent clauses. The UML classes modelling these data structures are included in the complete UML model in Annex C.

Each of the SAS operations is described in more detail in subsequent clauses.

7 Shared aspects

7.1 Introduction

This clause specifies aspects of the SAS Service behavior that are shared by several operations.

7.2 Shared operation parameters

This clause specifies some of the parameters used by multiple operations specified in the following clauses. The parameter names, meanings, data types, and multiplicity shall be as specified in Table 1.

Table 1 — Definitions of some operation request and response parameters

Name	Definition	Data type and value
AlertFormat	Unique identifier or definition of format used to encode alerts	ows:MimeType
AlertTime	Alert time periods that this publisher will publish. This adds a way to specify the publication of historical alerts.	ogc:temporalOps
DesiredPublicationExpe ration	The time that the requestor would like the publication channel to expire. The server may grant a shorter validity time than requested.	gml:TimeInstant
FeatureOfInterest	Specifies target feature for which alerts are published. Mostly a helper for in-situ sensors, since geo-location has to be done on the server side. The supported area should be listed in the selected offering capabilities.	
ObservedProperty	Observables that are associated with the alerts.	anyURI
Procedure	Reference to a particular sensor system advertised in the offering. Allows client to request alerts from one or more sensors in the offering. This is to support scenarios with sensor grids (we don't want to have one offering for each sensor in that case). Note that sensorML can describe Sensor Arrays too.	om:Procedure
PublicationEndpoint	A service endpoint reference that is intended to be accessed by the publisher to publish alerts. Typically this will be a URL. If the protocol that is being used is XMPP the URL will be of the form xmpp://jid@host according to IETF draft draft-saintandre-xmpp-iri-01. This allows a client to connect to a server to publish [jst6]notifications rather than to require the client to accept connections from the server. This should be returned when the client has not sent a ClientEndpoint reference to the server.	URL

Name	Definition	Data type and value
PublicationID	ID administered by SAS server to uniquely identify advertisement offerings.	ID
PublicationExpiration	The time that this publication channel automatically terminates.	gml:TimeInstant
Result	This element is a container to hold the value of some Observation in O&M since that schema does not have such an element. It is used to construct filter expressions that represent alert conditions.[jst7]	anyType[AW8]
ResultDefinition	Alerts are reported only when the result of an associated observation match this expression.	ogc:ComparisonOps
ResultFormat	Defines the mime type of the result. Should be of type xml usually.	ows:MimeType
SubscriptionEndpoint	A service endpoint reference that is intended to be accessed by the requestor to receive the notification messages as a result of this subscription. Typically this will be a URL. If the protocol being used is XMPP the url will be of the form xmpp://jid@host according to IETF draft draft-saintandre-xmpp-iri-01. This allows a client to connect to a server to receive notifications rather than to require the client to accept connections from the server for posting notifications. This should be returned when the client has not sent a ClientEndpoint reference to the server.	wsa:EndpointReferenceType
SubscriptionOfferingID	This unique identifier identifies the SubscriptionOffering. It is part of the Capabilities document and can be used by clients in Subscribe operation requests.	token
SubscriptionID	Unique Identifier for the subscription. Provided by SAS server.	ID
SubscriptionExpiration	The time that this subscription automatically terminates.	gml:TimeInstant
SystemDescriptionEndpoint	A service endpoint reference that provides access to a system description document	wsa:EndpointReferenceType
TransportBinding	The URI represents the definition for the given transport. These are expected to be defined in a GML dictionary. Sampel values are: urn:ogc:def:transportbinding:xmpp:muc and urn:ogc:def:transportbinding:http:post	URI

Name	Definition	Data type and value

7.3 Operation request encoding

The encoding of operation requests shall use HTTP GET with KVP encoding and HTTP POST with XML encoding as specified in Clause 11 of [OGC 05-008]. Table 2 summarizes the SAS Service operations and their encoding methods defined in this specification.

Table 2 — Operation request encoding

Operation name	Request encoding
GetCapabilities (required)	KVP and optional XML
All others	XML

8 GetCapabilities operation (mandatory)

8.1 Introduction

The mandatory GetCapabilities operation allows clients to retrieve service metadata from a server. The response to a GetCapabilities request shall be an XML document containing service metadata about the server, including specific information about SAS. This clause specifies the XML document that a SAS server must return to describe its capabilities.

8.2 Operation request

The GetCapabilities operation request shall be as specified in Subclauses 7.2 and 7.3 of [OGC 05-008]. The value of the "service" parameter shall be "SAS". The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Tables 3 and 7 of [OGC 05-008], with the additions listed in Table 3 below.

Table 3 — Additional Section name values and meanings

Section name	Meaning
sas:Contents	Return the contents section in service metadata document that contains information about the subscription offerings and advertisement offerings.

The "Multiplicity and use" column in Table 1 of [OGC 05-008] specifies the optionality of each listed parameter in the GetCapabilities operation request. Table 4 specifies the implementation of those parameters by SAS clients and servers.

Table 4 — Implementation of parameters in GetCapabilities operation request

Name	Multiplicity	Client implementation	Server implementation
service	One (mandatory)	Each parameter shall be implemented by all clients, using specified values	Each parameter shall be implemented by all servers, checking that each parameter is received with specified values
request	One (mandatory)		
AcceptVersions	Zero or one (optional)	Should be implemented by all software clients, using specified values	Shall be implemented by all servers, checking if parameter is received with specified value(s)
Sections	Zero or one (optional) [b]	Each parameter may be implemented by each client [b] If parameter not provided, shall expect default response If parameter provided, shall allow default or specified response	Each parameter may be implemented by each server [a] If parameter not implemented or not received, shall provide default response If parameter implemented and received, shall provide specified response
updateSequence	Zero or one (optional) [b]		
AcceptFormats	Zero or one (optional) [b]		

a A specific OWS is allowed to make mandatory server implementation of any of these three parameters.

b If a specific OWS makes mandatory server implementation of any of these three parameters, that parameter can also be made mandatory in the operation request, also requiring client implementation of this parameter.

All SAS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only.

EXAMPLE 1 To request a SAS capabilities document, a client could issue the following KVP encoded GetCapabilities operation request with near-minimum contents:

`http://mars.uni-muenster.de/SAS/SAS?Request=GetCapabilities&Service=SAS`

EXAMPLE 2 The corresponding GetCapabilities operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/sas" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk/sas/0.0.1/sasGetCapabilities.xsd" service="SAS">
</GetCapabilities>
```

8.3 GetCapabilities operation response

8.3.1 Normal response

The service metadata document shall contain the SAS sections specified in Table 5. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections can be requested and shall be returned when requested.

Table 5 — Section name values and contents

Section name	Contents
ServiceIdentification	Metadata about this specific server. The schema of this section shall be the same as for all OWSs, as specified in Subclause 7.4.3 and owsServiceIdentification.xsd of [OGC 05-008].
ServiceProvider	Metadata about the organization operating this server. The schema of this section shall be the same for all OWSs, as specified in Subclause 7.4.4 and owsServiceProvider.xsd of [OGC 05-008].
OperationsMetadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The basic contents and organization of this section shall be the same as for all OWSs, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008].
Contents	Metadata about the data served by this server. For the SAS, this section shall contain data about subscriptions and advertisements offered at this SAS interface, as specified in following Subclauses below.
Filter_Capabilities	Optional section containing Metadata about the supported filter mechanisms.

In addition to these sections, each service metadata document shall include the mandatory "version" and optional "updateSequence" parameters specified in Table 6 in Subclause 7.4.1 of [OGC 05-008].

8.3.2 OperationsMetadata section standard contents

For the SAS, the OperationsMetadata section shall be the same as for all OGC Web Services, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008]. The mandatory values of various (XML) attributes shall be as specified in Table 6. Similarly, the optional attribute values listed in **Error! Reference source not found.** shall be included or not depending on whether that operation is implemented by that server. In Table 6 and **Error! Reference source not found.**, the "Attribute name" column uses dot-separator notation to identify parts of a parent item. The "Attribute value**"** column references an operation parameter, in this case an operation name, and the meaning of including that value is listed in the right column.

Table 6 — Required values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	GetCapabilities	The operation is implemented by this server.
	Advertise	The operation is implemented by this server.
	CancelAdvertisement	The operation is implemented by this server.
	RenewAdvertisement	The operation is implemented by this server.
	Subscribe	The operation is implemented by this server.

	CancelSubscription	The operation is implemented by this server.
	RenewSubscription	The operation is implemented by this server.

8.3.3 Contents section

The Contents section of a service metadata document contains metadata about the data served by this server. For the SAS, this Contents section shall contain data about subscriptions and advertisements: The SAS interface allows clients to subscribe to alerts or to publish alerts. The two necessary sections provide all information a client needs to perform its intended task.

In case that a client wants to advertise alerts, the AdvertisementOffering Element defines syntax and structure how the offering has to be encoded. The client has to[AW9] provide information about the following questions:

- What is the observed property and where is the observation located?
- Which alert format will be used (e.g. TML, CAP, …)?
- Which transport binding is used by the associated messaging server (e.g. XMPP)?

The following figure shows the AdvertisementOffering element in UML notation.

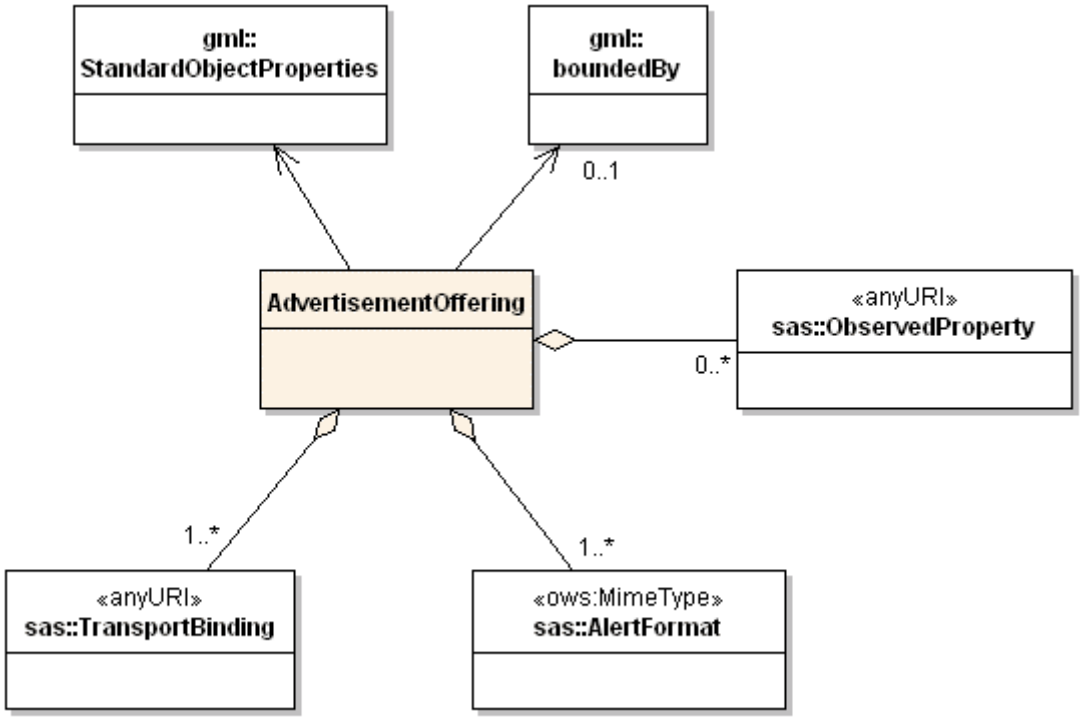


Figure 6: AdvertisementOffering in UML notation

In case that a client wants to subscribe to alerts, the SubscriptionOffering Element contains all information how the final (at the messaging server) subscription has to be performed. The SubscriptionOffering is shown in the following figure.

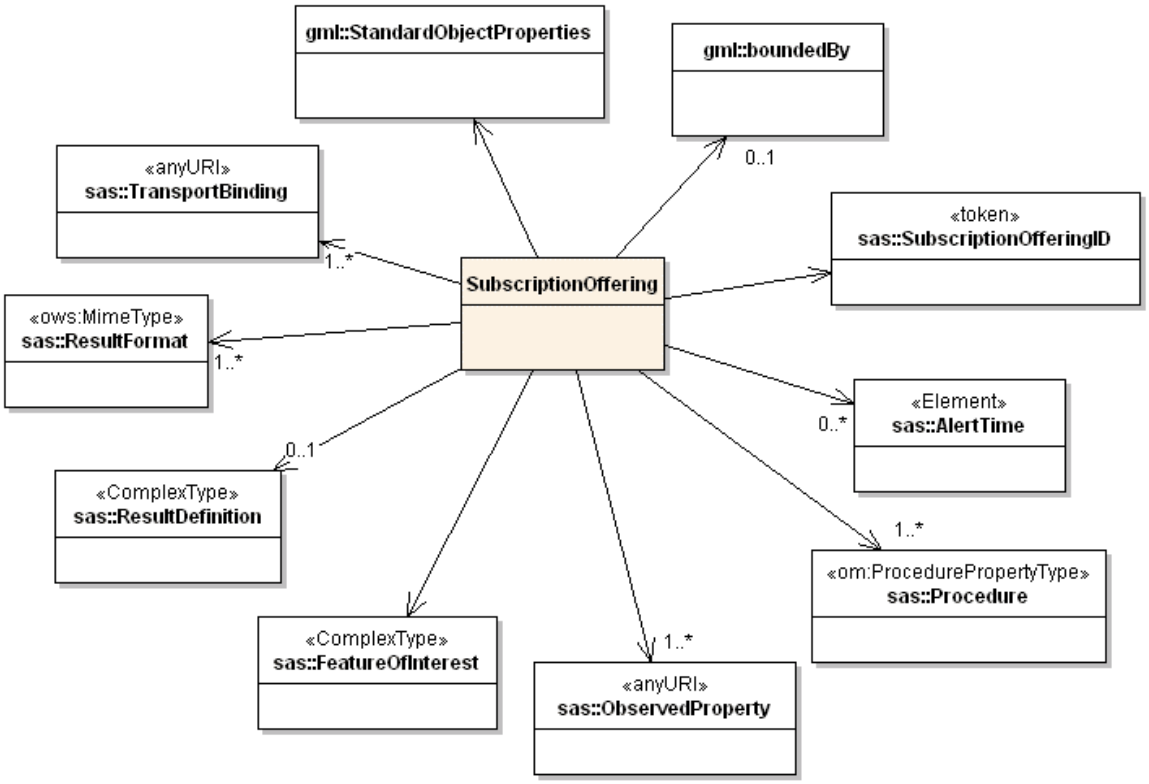


Figure 7: SubscriptionOffering element in UML notation

8.3.4 Capabilities document XML encoding[AW10]

A XML schema fragment for a SAS service metadata document extends ows:CapabilitiesBaseType in owsCommon.xsd of [OGC 05-008], and can be found in annex B (sasGetCapabilities.xsd).

8.3.5 Capabilities document example

In response to GetCapabilities operation request, a SAS server might generate a document that looks like (See Annex D1 for an entire capabilities document):

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/sas" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengeospatial.net/ows" xmlns:om="http://www.opengis.net/om"
xmlns:gml="http://www.opengis.net/gml" xmlns:swe="http://www.opengis.net/swe"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/sas
http://mars.uni-muenster.de/swerep/trunk/sas/0.0.1/sasGetCapabilities.xsd" version="0.0.1">
    <Contents>
        <SubscriptionOfferingList>
            <SubscriptionOffering>
                <gml:boundedBy>
                    <gml:Envelope srsName="EPSG:4326">
                        <gml:lowerCorner>7.727957 51.883905</gml:lowerCorner>
                        <gml:upperCorner>7.727959 51.883907</gml:upperCorner>
                    </gml:Envelope>
```

```
                </gml:boundedBy>
                <SubscriptionOfferingID>subOf-1</SubscriptionOfferingID>
                <Procedure xlink:href="system.ifgi-sensor-1"/>
                <:ObservedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel"/>
                <FeatureOfInterest>
                    <om:Station>
                        <gml:name>ALBER</gml:name>
                        <om:position>
                            <gml:Point>
                                <gml:pos srsName="urn:ogc:crs:epsg4326">7.727958 51.883906</gml:pos>
                            </gml:Point>
                        </om:position>
                        <om:procedureHosted xlink:href="system.ifgi-sensor-1"/>
                    </om:Station>
                </FeatureOfInterest>
                <ResultDefinition>
                    <ogc:PropertyIsGreaterThan>
                        <ogc:Literal>130</ogc:Literal>
                    </ogc:PropertyIsGreaterThan>
                </ResultDefinition>
                <ResultFormat>text/xml;subtype=&quot;tml/2.0&quot;</ResultFormat>
                <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
            </SubscriptionOffering>
        </SubscriptionOfferingList>
        <AdvertisementOfferingList>
            <AdvertisementOffering gml:id="pub-1">
                <gml:boundedBy>
                    <gml:Envelope srsName="EPSG:4326">
                        <gml:lowerCorner>7.727957 51.883905</gml:lowerCorner>
                        <gml:upperCorner>7.727959 51.883907</gml:upperCorner>
                    </gml:Envelope>
                </gml:boundedBy>
                <ObservedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel"/>
                <AlertFormat>text/xml;subtype=&quot;tml/2.0&quot;</AlertFormat>
                <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
            </AdvertisementOffering>
        </AdvertisementOfferingList>
    </Contents>
</Capabilities>
```

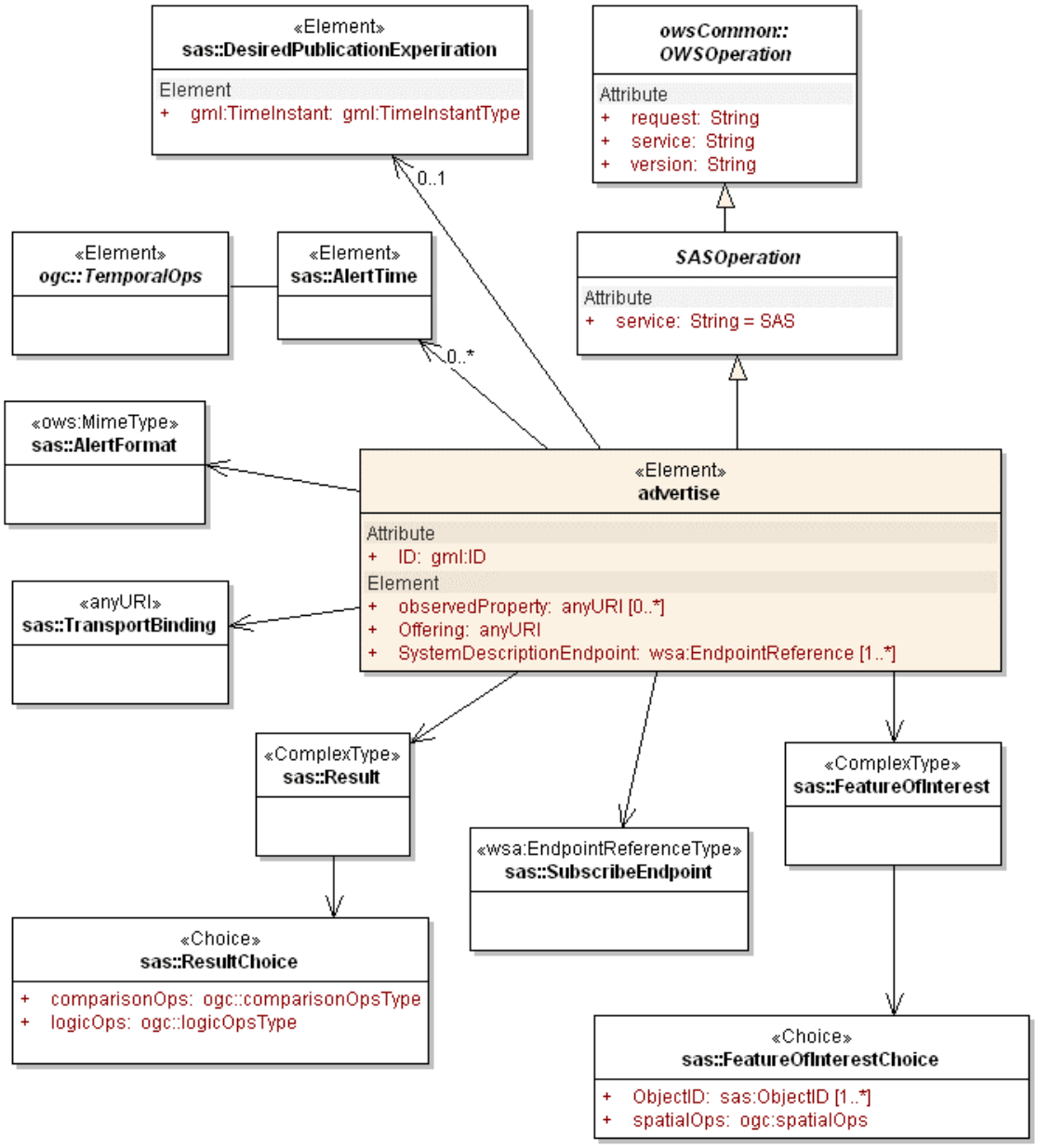
8.3.6 Exceptions

When a SAS server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 05-008]. The allowed exception codes shall include those listed in Table 5 of Subclause 7.4.1 of [OGC 05-008], if the updateSequence parameter is implemented by the server.

9 Advertise operation (mandatory)

9.1 Introduction

The advertise operation allows SAS clients to advertise what kind of data could be published. The operation is mainly used by sensor management units that want to register the sensor at the SAS[AW11].



[AW12]

Figure 8: Advertise element in UML notation

9.2 Advertise operation request

9.2.1 Advertise request parameters

This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Table 7 — Parameters in Advertise operation request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
SystemDescriptionEndpoint	see Subclause 7.2	Wsa:EndpointReferenceType	One to many (mandatory)
Result	see Subclause 7.2	ogc:comparisonOps or ogc:logicOps	One (mandatory)
AlertFormat	see Subclause 7.2	ows:MimeType	One (mandatory)
AdvertisementID	ID of an offering that might be advertised in the capabilities document as it is if unambiguousness has been insured.	anyURI type	optional
AlertTime	see Subclause 7.2	ogc:temporalOps	One to many (optional)
ObservedProperty	see Subclause 7.2	anyURI type	One to many (optional)
FeatureOfInterest	see Subclause 7.2	ogc:spatialOps or ObjectID	optional
DesiredPublicationExpiration	see Subclause 7.2	gml:TimeInstant	optional
TransportBinding	see Subclause 7.2	anyURItype	optional
SubscribeEndpoint	see Subclause 7.2	wsa:EndpointReferenceType	optional
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

NOTE 2 The data type of many parameters is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.[jst13]

The "Multiplicity and use" columns in all tables specify the optionality of each listed parameter and data structure in the Advertise operation request. All the "mandatory" parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all the "mandatory" parameters and data structures shall be implemented by all SAS servers, checking that each request parameter or data structure is received with any specified value(s).

All the "optional" parameters and data structures, in the Advertise operation request, should be implemented by all SAS clients using specified values, for each implemented

process to which that parameter or data structure applies. Similarly, all the “optional” parameters and data structures shall be implemented by all SAS servers, for each implemented process to which that parameter or data structure applies.

9.2.2 Advertise request KVP encoding

KVP encoding is not supported.

9.2.3 Advertise request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the Advertise operation request, using XML encoding only. The following schema fragment specifies the contents and structure of an Advertise operation request encoded in XML:

See annex B (sasAdvertise.xsd).

EXAMPLE An example Advertise operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Advertise xmlns="http://www.opengis.net/sas" xmlns:sos="http://www.opengeospatial.net/sos"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk/sas/0.0.1/sasAdvertise.xsd" service="SAS" version="0.0.1">
    <SystemDescriptionEndpoint>
        <wsa:Address>http://mars.uni-muenster.de:8080/OWS3-SOS/sos</wsa:Address>
        <wsa:ReferenceParameters>
            <sos:DescribeSensor service="SOS"
outputFormat="text/xml;subtype=&quot;sensorML/1.0.0&quot;" version="0.0.31">
                <SensorId>system.ifgi-sensor-1</SensorId>
            </sos:DescribeSensor>
            <sml:outputs>
                <sml:OutputList>
                    <sml:output name="urn:ogc:def:phenomenon:OGC:1.0.31:waterlevel"/>
                </sml:OutputList>
            </sml:outputs>
        </wsa:ReferenceParameters>
        <wsa:ServiceName>sos:DescribeSensor</wsa:ServiceName>
    </SystemDescriptionEndpoint>
    <Result>
        <ogc:PropertyIsGreaterThan>
            <ogc:Literal>130</ogc:Literal>
        </ogc:PropertyIsGreaterThan>
    </Result>
    <AlertFormat>text/xml;subtype=&quot;tml/2.0&quot;</AlertFormat>
    <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
</Advertise>
```

9.3 Advertise operation response

9.3.1 Normal response parameters

The normal response to a valid Advertise operation request shall be an AdvertiseResponse. More precisely, a response from the Advertise operation shall include the parts shown in the following figure. Note that all elements and attributes are explained in Subclause 7.2, shared aspects.

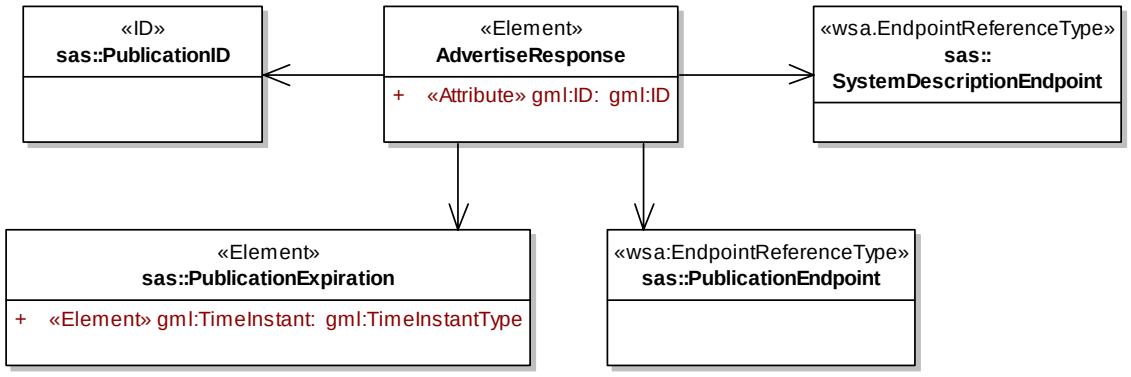


Figure 9: AdvertiseResponse in UML notation

9.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of an Advertise operation response, always encoded in XML.

See Annex B (sasAdvertise.xsd).

9.3.3 Advertise response example

```
<?xml version="1.0" encoding="UTF-8"?>
<AdvertiseResponse xmlns="http://www.opengis.net/sas" xmlns:gml="http://www.opengis.net/gml"
xmlns:sos="http://www.opengeospatial.net/sos" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk/sas/0.0.1/sasAdvertise.xsd" gml:id="advertiseRequest000001">
    <PublicationID>pub-1</PublicationID>
    <PublicationExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-02T18:00:00</gml:timePosition>
        </gml:TimeInstant>
    </PublicationExpiration>
    <PublicationEndpoint>
        <wsa:Address>xmpp://ifgi-1incoming@mars.uni-muenster.de</wsa:Address>
    </PublicationEndpoint>
    <SystemDescriptionEndpoint>
        <wsa:Address>http://mars.uni-muenster.de:8080/OWS3-SOS/sos</wsa:Address>
        <wsa:ReferenceParameters>
            <sos:DescribeSensor service="SOS"
outputFormat="text/xml;subtype=&amp;quot;sensorML/1.0.0&amp;quot;" version="0.0.31">
                <SensorId>system.ifgi-sensor-1</SensorId>
            </sos:DescribeSensor>
            <sml:outputs>
                <sml:OutputList>
                    <sml:output name="urn:ogc:def:phenomenon:OGC:1.0.31:waterlevel"/>
                </sml:OutputList>
            </sml:outputs>
        </wsa:ReferenceParameters>
        <wsa:ServiceName>sos:DescribeSensor</wsa:ServiceName>
    </SystemDescriptionEndpoint>
</AdvertiseResponse>
```

9.3.4 Advertise exceptions

When a SAS server encounters an error while performing an Advertise operation, it shall return an exception report message as specified in clause 8 of[AW14] [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 8. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 8.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 8 — Exception codes for Advertise operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter[jst15][IS16] [AW17]

10 RenewAdvertisement operation (mandatory)

10.1 Introduction

The RenewAdvertisement operation allows SAS clients to renew a previously advertised advertisement. The operation is mainly used by sensor management units that had registered a sensor at the SAS but the advertisement time that was set by either the sensor management unit itself or the SAS has expired.

10.2 RenewAdvertisement operation request

10.2.1 RenewAdvertisement request parameters

A request to perform the RenewAdvertisement operation shall include the parameters listed and defined inTable 9. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the "Name" column appear to contain spaces, they shall not contain spaces.

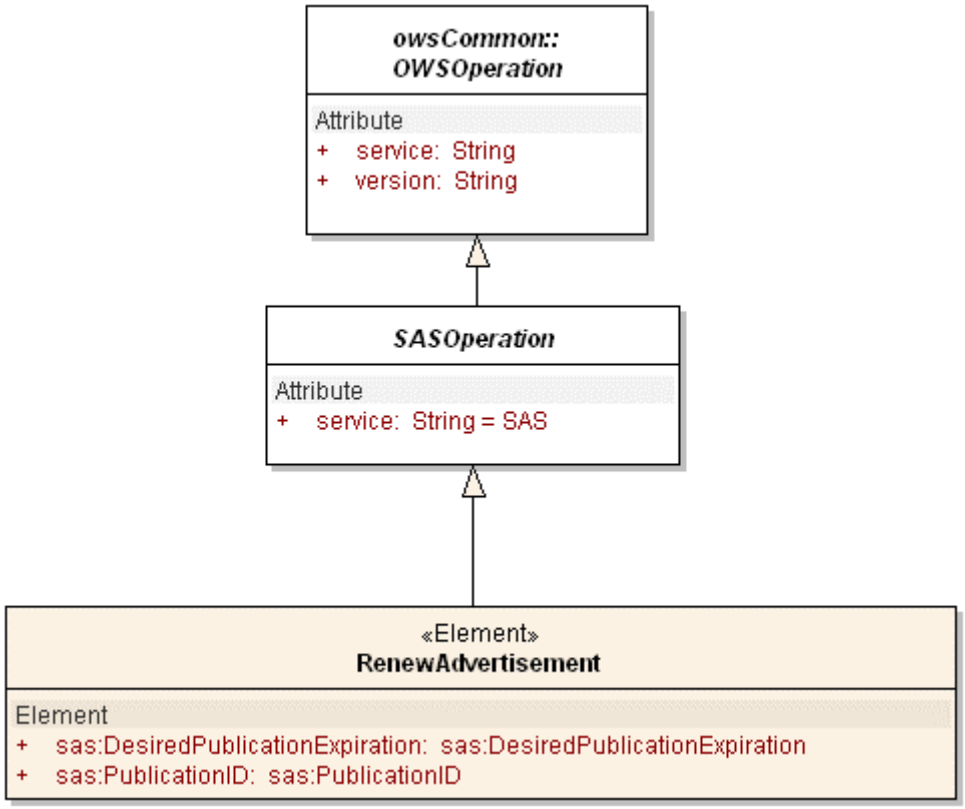


Figure 10: RenewAdvertisement operation in UML notation

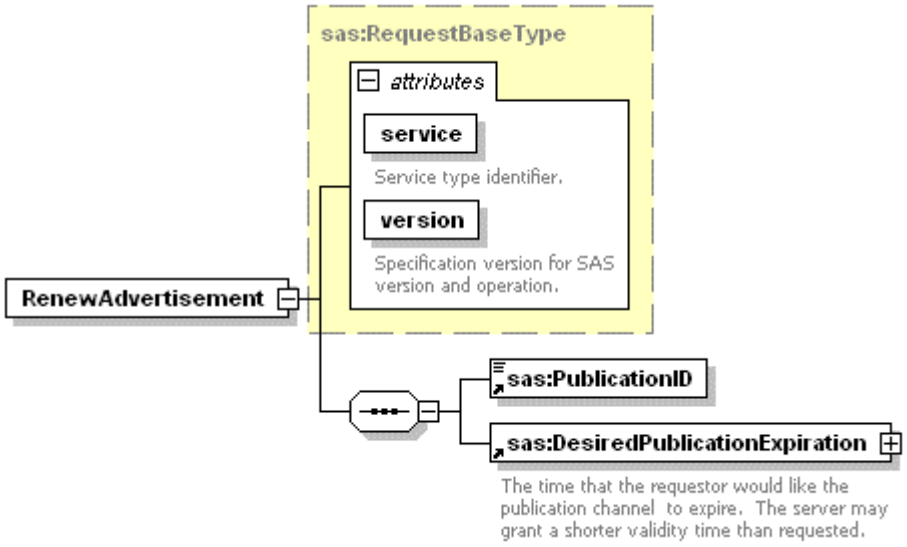


Figure 11: RenewAdvertisement operation in XMLSpy notation

Table 9 — Parameters in RenewAdvertisement operation request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., “WMS”, “WFS”)	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
PublicationID	ID administered by SAS server to uniquely identify advertisement offerings.	ID	One (mandatory)
DesiredPublicationExpiration	Defines the time this advertisement shall expire	gml:TimeInstant	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

NOTE 2 The data type of many parameters is specified as “Character String type, not empty”. In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

The “Multiplicity and use” column in Table 9 specifies the optionality of each listed parameter and data structure in the RenewAdvertisement operation request. Since all parameters and data structures are mandatory in the operation request, all parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all parameters and data structures shall be implemented by all SAS servers, checking that each request parameter is received with any specified value(s).

10.2.2 RenewAdvertisement request KVP encoding

KVP encoding not supported!

10.2.3 RenewAdvertisement request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the RenewAdvertisement operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a RenewAdvertisement operation request encoded in XML:

See annex B (sasAdvertise.xsd).

EXAMPLE An example RenewAdvertisement operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<RenewAdvertisement xmlns="http://www.opengis.net/sas" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:gml="http://www.opengis.net/gml" xmlns:ns1="http://www.opengeospatial.net/sos"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
```

```
http://mars.uni-muenster.de/swerep/trunk/SAS/0.0.1/sasAdvertise.xsd" service="SAS" version="0.0.1">
    <PublicationID>pub-1</PublicationID>
    <DesiredPublicationExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T08:00:00</gml:timePosition>
        </gml:TimeInstant>
    </DesiredPublicationExpiration>
</RenewAdvertisement>
```

10.3 RenewAdvertisement operation response

10.3.1 Normal response parameters

The normal response to a valid RenewAdvertisement operation request shall be RenewAdvertisementResponse. More precisely, a response from the RenewAdvertisement operation shall include the parts listed in Table 10. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

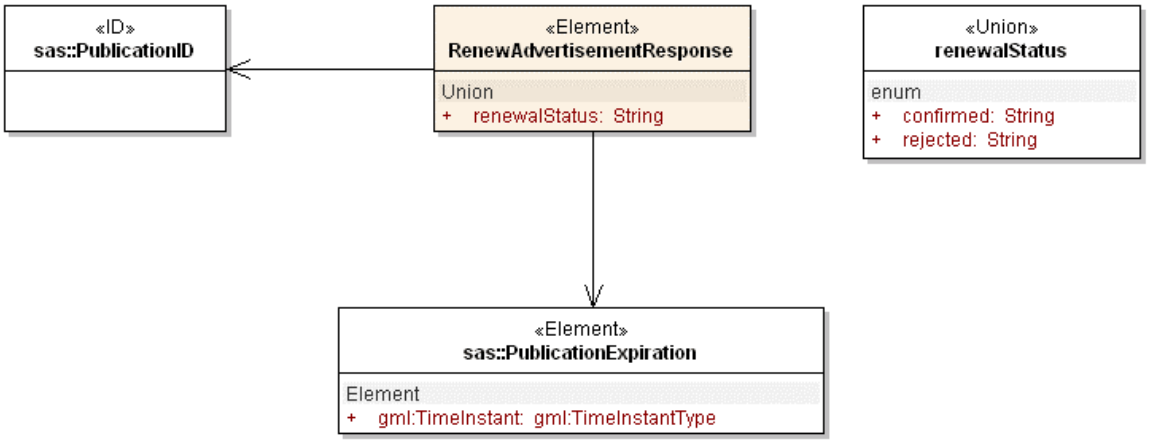


Figure 12: RenewAdvertisementResponse in UML notation

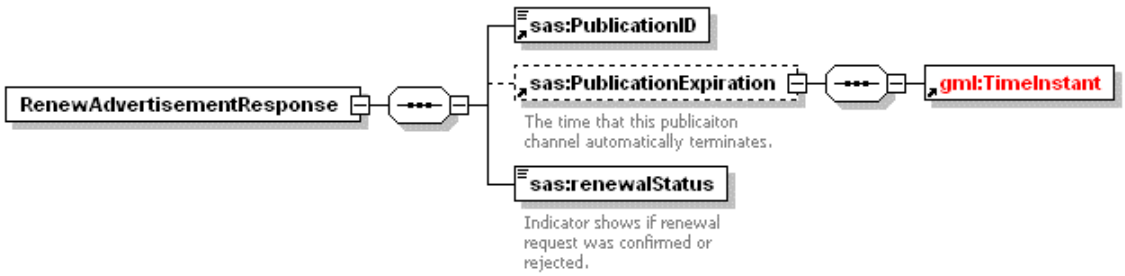


Figure 13: RenewAdvertisementResponse in XMLSpy notation

Table 10 — Parts of RenewAdvertisement operation response

Name	Definition	Data type and values	Multiplicity and use
PublicationID	ID administered by SAS server to uniquely identify advertisement offerings.	ID	one (mandatory)
PublicationExpiration	Defines the time this advertisement will expire.	gml:TimeInstant	optional
renewalStatus	Identifier if renewal was successful	String “confirmed” or “rejected”	one (mandatory)
a			

10.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of a RenewAdvertisement operation response, always encoded in XML:

See annex B (sasAdvertise.xsd).

10.3.3 RenewAdvertisement response example

A RenewAdvertisement operation response for SAS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<RenewAdvertisementResponse xmlns="http://www.opengis.net/sas"
xmlns:gml="http://www.opengis.net/gml" xmlns:sos="http://www.opengeospatial.net/sos"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk/SAS/0.0.1/sasAdvertise.xsd">
    <PublicationID>pub-1</PublicationID>
    <PublicationExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T07:00:00</gml:timePosition>
        </gml:TimeInstant>
    </PublicationExpiration>
    <renewalStatus>confirmed</renewalStatus>
</RenewAdvertisementResponse>
```

10.3.4 RenewAdvertisement exceptions

When a SAS server encounters an error while performing a RenewAdvertisement operation, it shall return an exception report message as specified in Subclause 7 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 11. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 11.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 11 — Exception codes for RenewAdvertisement operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

11 CancelAdvertisement operation (mandatory)

11.1 Introduction

The CancelAdvertisement operation allows SAS clients to cancel a previously registered advertisement offering. The operation is mainly used by sensor management units that want to unregister the sensor at the SAS. The only request payload is the PublicationID that was provided by the SAS server.

11.2 CancelAdvertisement operation request

11.2.1 CancelAdvertisement request parameters

A request to perform the CancelAdvertisement operation shall include the parameters listed and defined in Table 7. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the "Name" column appear to contain spaces, they shall not contain spaces.

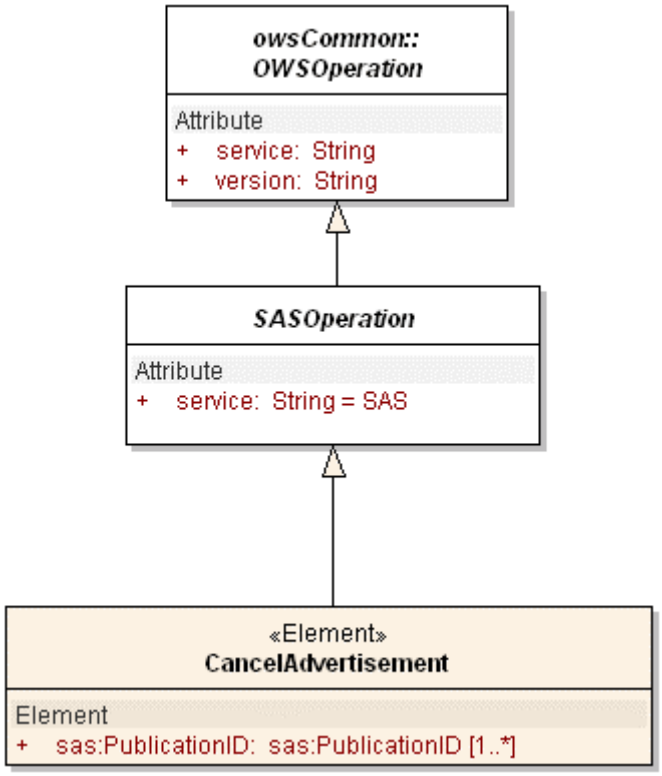


Figure 14: CancelAdvertisement in UML notation

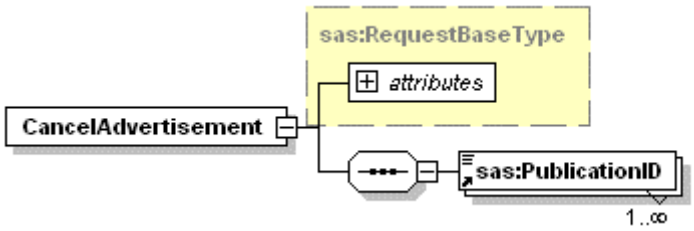


Figure 15: CancelAdvertisement in XMLSpy notation

Table 12 — Parameters in CancelAdvertisement operation request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
PublicationID	ID administered by SAS server to uniquely identify advertisement offerings.	ID	One to many [AW18]([jst19]mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

NOTE 2 The data type of many parameters is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

The "Multiplicity and use" column in Table 12 specifies the optionality of each listed parameter and data structure in the CancelAdvertisement operation request. Since all parameters and data structures are mandatory in the operation request, all parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all parameters and data structures shall be implemented by all SAS servers, checking that each request parameter is received with any specified value(s).

11.2.2 CancelAdvertisement request KVP encoding

KVP encoding not supported!

11.2.3 CancelAdvertisement request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the CancelAdvertisement operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a CancelAdvertisement operation request encoded in XML:

See annex B (sasAdvertisement.xsd).

EXAMPLE An example CancelAdvertisement operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<CancelAdvertisement service="SAS" version="0.0.1" xmlns="http://www.opengis.net/sas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk\SAS\0.0.1\sasAdvertise.xsd">
    <PublicationID>pub-1</PublicationID>
</CancelAdvertisement>
```

11.3 CancelAdvertisement operation response

11.3.1 Normal response parameters

The normal response to a valid CancelAdvertisement operation request shall be CancelAdvertisementResponse. More precisely, a response from the CancelAdvertisement operation shall include the parts listed in Table 13. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

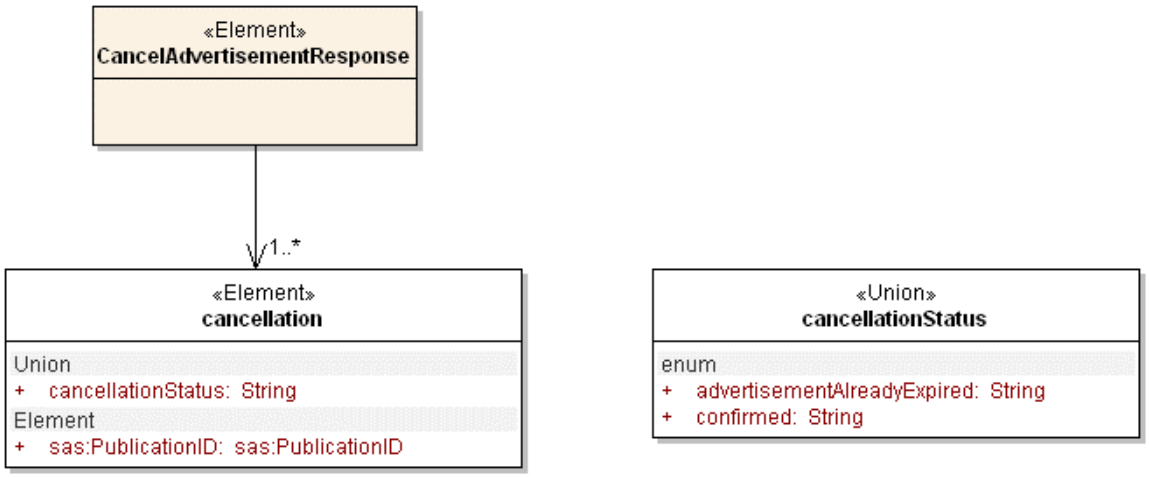


Figure 16CancelAdvertisementResponse in UML notation

Figure 17: CancelAdvertisementResponse

Table 13 — Parts of CancelAdvertisement operation response

Name	Definition	Data type and values	Multiplicity and use
cancellation	Element containing the following two subelements to provide information about the requested cancellation.	-	one to many (mandatory)
cancellation.PublicationID	see Subclause 7.2		one (mandatory)
cancellation.cancellationStatus	Identifies the status of the cancellation request	String “confirmed” “already expired”	one (mandatory)

11.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of a CancelAdvertisement operation response, always encoded in XML:

See Annex B (sasAdvertise.xsd).

11.3.3 CancelAdvertisement response example

A CancelAdvertisement operation response for SAS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<CancelAdvertisementResponse xmlns="http://www.opengis.net/sas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep/trunk/SAS/0.0.1/sasAdvertise.xsd">
    <cancellation>
        <PublicationID>pub-1</PublicationID>
        <cancellationStatus>confirmed</cancellationStatus>
    </cancellation>
</CancelAdvertisementResponse>
```

11.3.4 CancelAdvertisement exceptions

When a SAS server encounters an error while performing a CancelAdvertisement operation, it shall return an exception report message as specified in Subclause 8 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 14. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 14.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 14 — Exception codes for CancelAdvertisement operation

exceptionCode value	Meaning of code	“locator” value

OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

12 Subscribe operation (mandatory)

12.1 Introduction

The Subscribe operation allows SAS clients to subscribe to alerts that are advertised and published at this SAS. The consumer can define custom alerts, based on the advertised and published alerts by filtering.[jst20]

12.2 Subscribe operation request

12.2.1 Subscribe request parameters

A request to perform the Subscribe operation shall include the parameters listed and defined in Table 15. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the "Name" column appear to contain spaces, they shall not contain spaces.

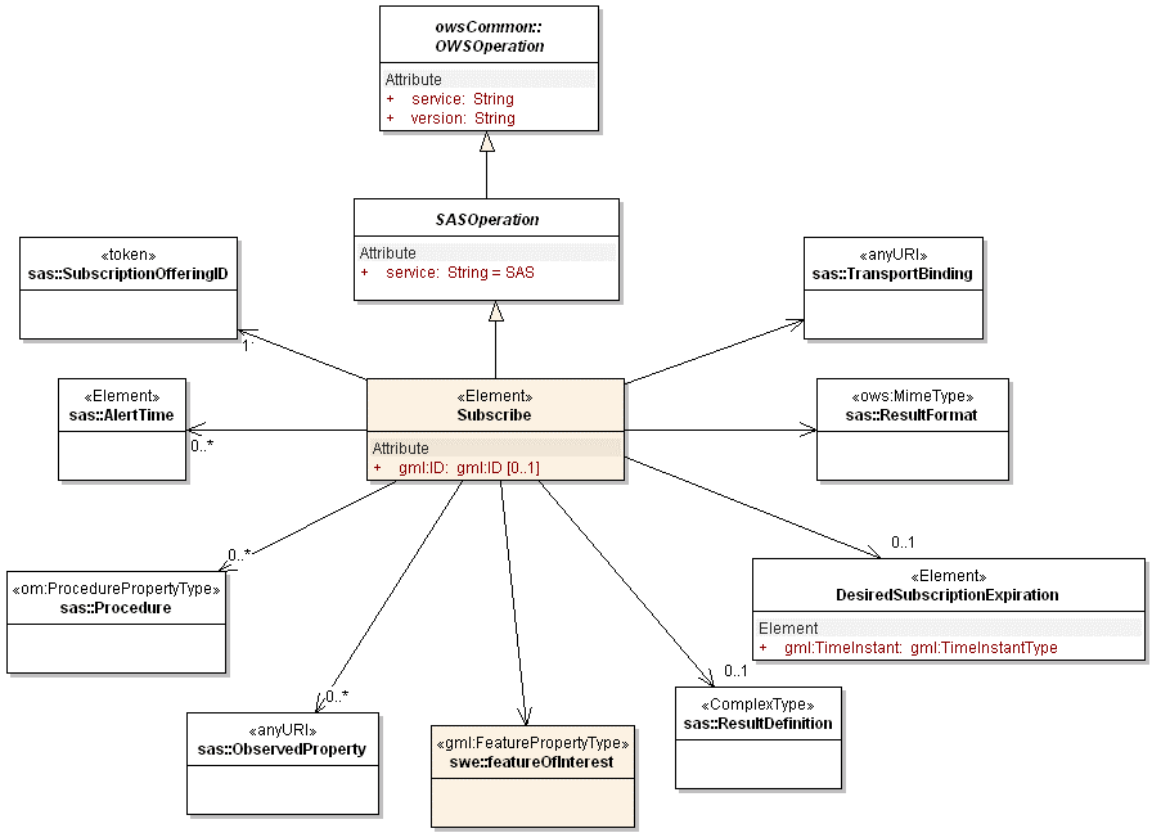


Figure 18: Subscribe in UML notation

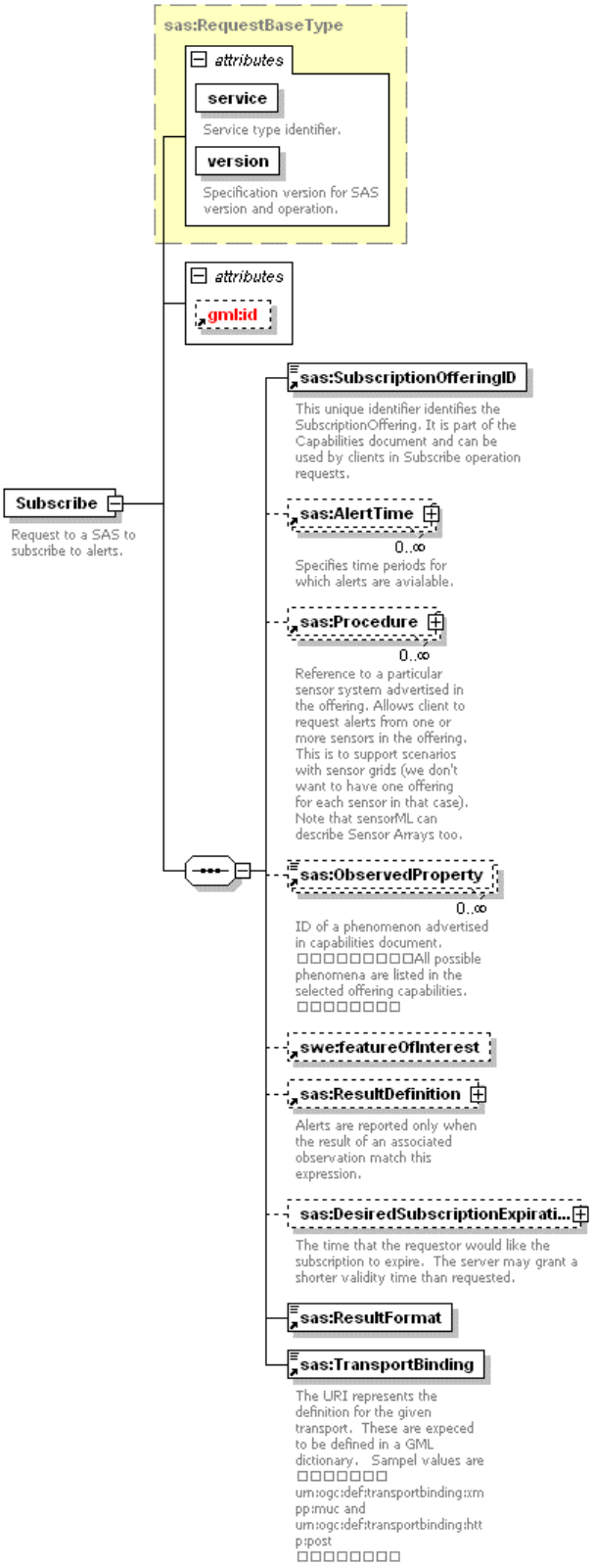


Figure 19: Subscribe in XMLSpy notation

Table 15 — Parameters in Subscribe operation request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
SubscriptionOfferingID	See Subclause 7.2	Token	One (mandatory)
ResultFormat	See Subclause 7.2	ows:MimeType	One (mandatory)
TransportBinding	See Subclause 7.2	anyURI type	One (mandatory)
AlertTime	See Subclause 7.2	ogc:temporalOps	One to many (optional)
Procedure	See Subclause 7.2	om:ProcedurePropertyType	One to many (optional)
ObservedProperty	See Subclause 7.2	anyURI type	One to many (optional)
FeatureOfInterest	See Subclause 7.2	gml:FeaturePropertyType[AW21]	optional
ResultDefinition	See Subclause 7.2	ogc:ComparisonOps	optional
DesiredSubscriptionExpiration[AW22]	The time that the requestor would like the subscription channel to expire. The server may grant a shorter validity time than requested.	gml:TimeInstant	optional

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

The "Multiplicity and use" column in Table 15 specifies the optionality of each listed parameter and data structure in the Subscribe operation request. All the "mandatory" parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all the "mandatory" parameters and data structures shall be implemented by all SAS servers, checking that each request parameter or data structure is received with any specified value(s).

All the "optional" parameters and data structures, in the Subscribe operation request, should be implemented by all SAS clients using specified values, for each implemented process to which that parameter or data structure applies. Similarly, all the "optional"

parameters and data structures shall be implemented by all SAS servers, for each implemented process to which that parameter or data structure applies.

12.2.2 Subscribe request KVP encoding

KVP encoding not supported!

12.2.3 Subscribe request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the Subscribe operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a Subscribe operation request encoded in XML:

See annex B (sasSubscribe.xsd).

EXAMPLE An example Subscribe operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscribe xmlns="http://www.opengis.net/sas" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:sos="http://www.opengis.net/sos" xmlns:om="http://www.opengis.net/om"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:swe="http://www.opengis.net/swe"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd" service="SAS" version="0.0.1">
    <SubscriptionOfferingID>subOf-1</SubscriptionOfferingID>
    <ObservedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</ObservedProperty>
    <swe:featureOfInterest>
        <om:Station>
            <gml:name>ALBER</gml:name>
            <om:position>
                <gml:Point>
                    <gml:pos srsName="urn:ogc:crs:epsg4326">7.727958 51.883906</gml:pos>
                </gml:Point>
            </om:position>
            <om:procedureHosted xlink:href="system.ifgi-sensor-1"/>
        </om:Station>
    </swe:featureOfInterest>
    <ResultDefinition>
        <ogc:And>
            <ogc:PropertyIsGreaterThan>
                <ogc:Literal>130</ogc:Literal>
            </ogc:PropertyIsGreaterThan>
            <ogc:PropertyIsLessThan>
                <ogc:Literal>150</ogc:Literal>
            </ogc:PropertyIsLessThan>
        </ogc:And>
    </ResultDefinition>
    <DesiredSubscriptionExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T06:00:00</gml:timePosition>
        </gml:TimeInstant>
    </DesiredSubscriptionExpiration>
    <ResultFormat>text/xml;subtype=&amp;quot;tml2.0&amp;quot;</ResultFormat>
    <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
</Subscribe>
```

12.3 Subscribe operation response

12.3.1 Normal response parameters

The normal response to a valid Subscribe operation request shall be SubscribeResponse. More precisely, a response from the Subscribe operation shall include the parts listed in Table 16. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

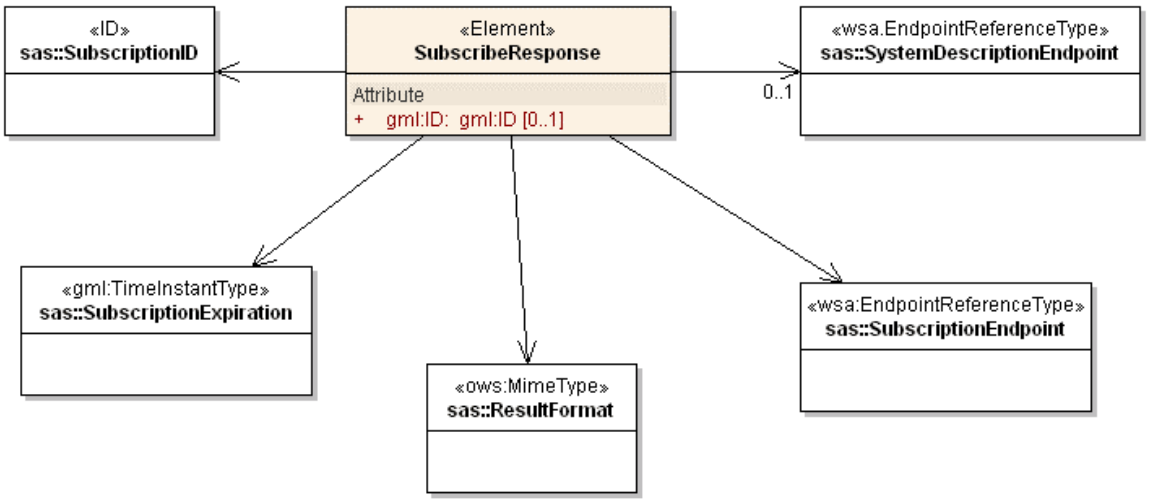


Figure 20: SubscribeResponse in UML notation

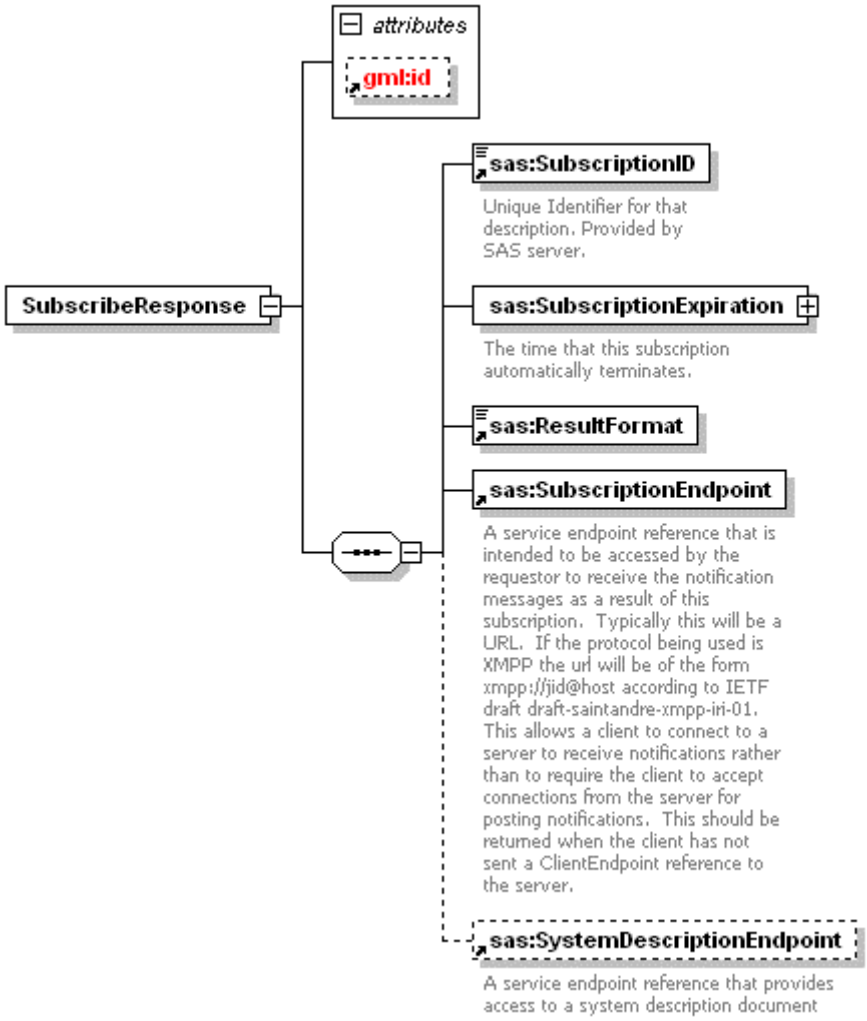


Figure 21: SubscribeResponse in XMLSpy notation

Table 16 — Parts of Subscribe operation response

Name	Definition	Data type and values	Multiplicity and use
SubscriptionID	Each subscription is identified by a unique identifier which is provided by the SAS.	ID	one (mandatory)
SubscriptionExpiration	The time that this subscription automatically terminates. Each SAS is free to choose the maximal time a subscription remains valid. After this period, the client has to send a SubscriptionRenewal request.	gml:Time	one (mandatory)
ResultFormat	Mime type of the result	Mime type	one (mandatory)
SubscriptionEndpoint	A service endpoint reference that is intended to be accessed by the requestor to receive the notification messages as a result of this subscription. Typically this will be a URL. If the protocol being used is XMPP the url will be of the form xmpp://jid@host according to IETF draft draft-saintandre-xmpp-iri-01. This allows a client to connect to a server to receive notifications rather than to require the client to accept connections from the server for posting notifications. This should be returned when the client has not sent a ClientEndpoint reference to the server.	wsa:EndpointReferenceType	one (mandatory)
SystemDescriptionEndpoint	A service endpoint reference that provides access to a system description document	wsa:EndpointReferenceType	optional
gml:id	TBD	ID	optional
a			

12.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of a Subscribe operation response, always encoded in XML:

See Annex B (sasSubscribe.xsd).

12.3.3 Subscribe response example

A Subscribe operation response for SAS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscribeResponse xmlns="http://www.opengis.net/sas" xmlns:sos="http://www.opengis.net/sos"
xmlns:gml="http://www.opengis.net/gml" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd">
    <SubscriptionID>sub-1</SubscriptionID>
    <SubscriptionExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T06:00:00</gml:timePosition>
        </gml:TimeInstant>
    </SubscriptionExpiration>
    <ResultFormat>text/xml;subtype=&quot;tml2.0&quot;</ResultFormat>
    <SubscriptionEndpoint>
        <wsa:Address>xmpp://sub-1@mars.uni-muenster.de</wsa:Address>
    </SubscriptionEndpoint>
    <SystemDescriptionEndpoint>
        <wsa:Address>http://mars.uni-muenster.de:8080/OWS3-SOS/sos</wsa:Address>
        <wsa:ReferenceParameters>
            <sos:DescribeSensor service="SOS"
outputFormat="text/xml;subtype=&quot;sensorML/1.0.0&quot;" version="0.0.31">
                <SensorId>system.ifgi-sensor-1</SensorId>
            </sos:DescribeSensor>
            <sml:outputs>
                <sml:OutputList>
                    <sml:output name="urn:ogc:def:phenomenon:OGC:1.0.31:waterlevel"/>
                </sml:OutputList>
            </sml:outputs>
        </wsa:ReferenceParameters>
        <wsa:ServiceName>sos:DescribeSensor</wsa:ServiceName>
    </SystemDescriptionEndpoint>
</SubscribeResponse>.
```

12.3.4 Subscribe exceptions

When a SAS server encounters an error while performing a Subscribe operation, it shall return an exception report message as specified in Subclause 8 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 17. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 17.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 17 — Exception codes for Subscribe operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

13 RenewSubscription operation (mandatory)

13.1 Introduction

The RenewSubscription operation allows SAS clients to renew the subscription that has expired. This operation becomes necessary as SAS servers determine the maximal time before a subscription expires automatically[AW23].[jst24]

13.2 RenewSubscription operation request

13.2.1 RenewSubscription request parameters

A request to perform the RenewSubscription operation shall include the parameters listed and defined in Table 18. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

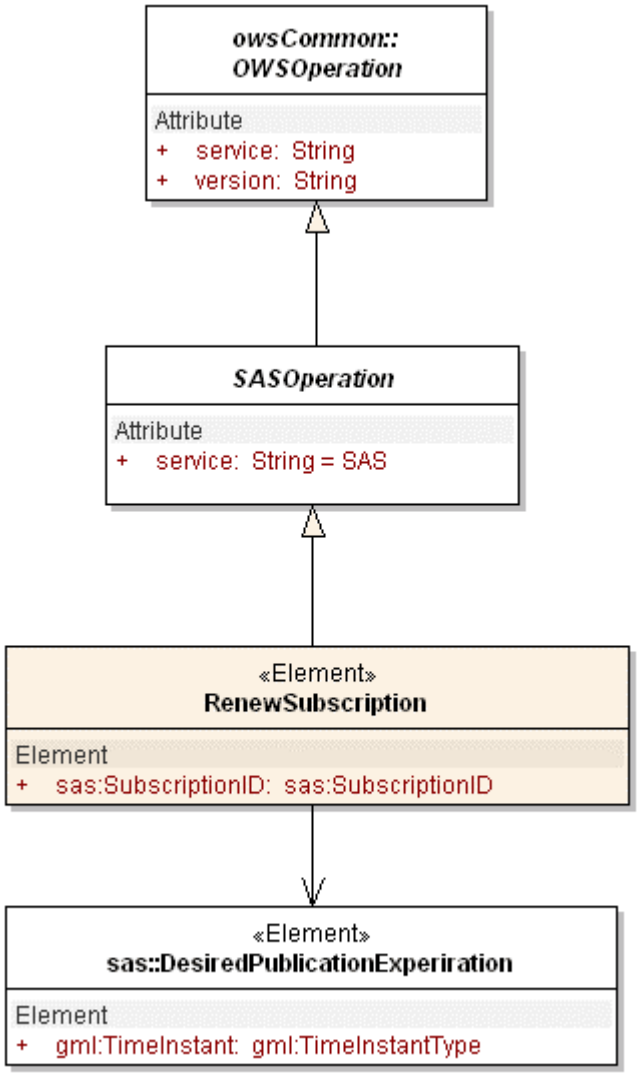


Figure 22: RenewSubscription in UML notation

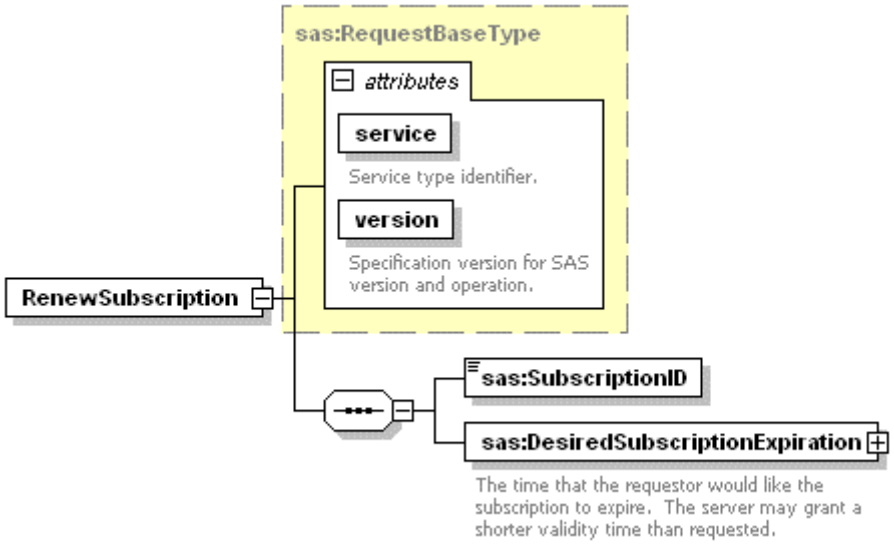


Figure 23: RenewSubscription in XMLSpy notation

Table 18 — Parameters in RenewSubscription operation request

Name a	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., “WMS”, “WFS”)	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
Subscription ID	Identifier provided by SAS as part of a SubscribeResponse (see Subclause 12)	ID	one (mandatory)
DesiredSubscriptionExpiration	Defines the time the client wants this subscription to expire. The final time will be determined by the server.	gml:TimeInstant	one (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

The “Multiplicity and use” column in Table 18 specifies the optionality of each listed parameter and data structure in the RenewSubscription operation request. Since all parameters and data structures are mandatory in the operation request, all parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all parameters and data structures shall be implemented by all SAS servers, checking that each request parameter is received with any specified value(s).

13.2.2 RenewSubscription request KVP encoding

KVP encoding not supported!

13.2.3 RenewSubscription request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the RenewSubscription operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a SAS operation request encoded in XML:

See annex B (sasSubscribe.xsd).

EXAMPLE An example RenewSubscription operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<RenewSubscription xmlns="http://www.opengis.net/sas" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd" service="SAS" version="0.0.1">
    <SubscriptionID>sub-1</SubscriptionID>
    <DesiredSubscriptionExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T08:45:00</gml:timePosition>
        </gml:TimeInstant>
    </DesiredSubscriptionExpiration>
</RenewSubscription>
```

13.3 RenewSubscription operation response

13.3.1 Normal response parameters

The normal response to a valid RenewSubscription operation request shall be RenewSubscriptionResponse. More precisely, a response from the RenewSubscription operation shall include the parts listed in Table 19. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

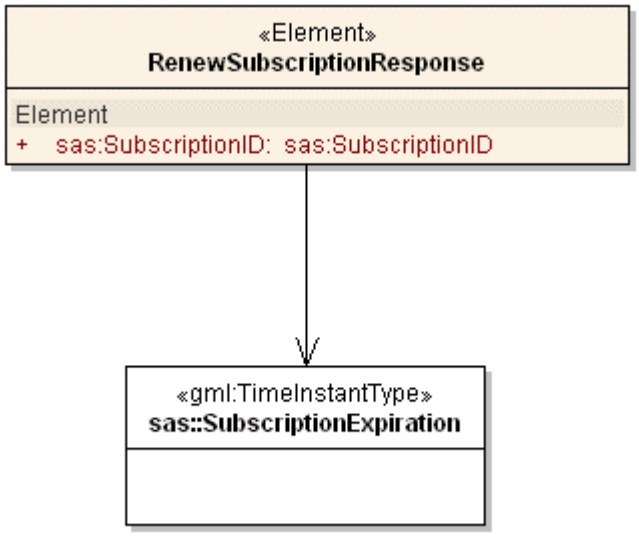


Figure 24: RenewSubscriptionResponse in UML notation

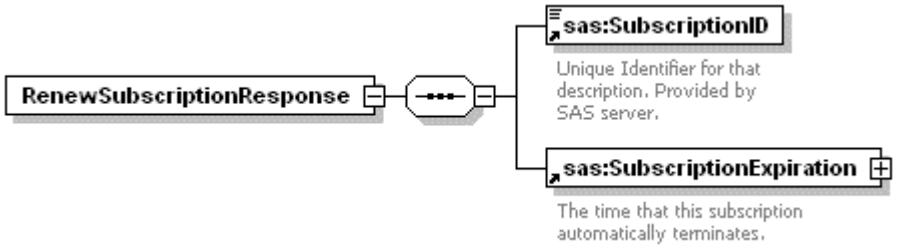


Figure 25: RenewSubscriptionResponse in XMLSpy notation

Table 19 — Parts of RenewSubscription operation response

Name	Definition	Data type and values	Multiplicity and use
SubscriptionID	Identifier provided by SAS as part of a SubscribeResponse (see Subclause 12)	ID	one (mandatory)
SubscriptionExpiration	The time that this subscription automatically terminates. Each SAS is free to choose the maximal time a subscription remains valid. After this period, the client has to send a SubscriptionRenewal request.	gml:Time	one (mandatory)

13.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of a RenewSubscription operation response, always encoded in XML:

See annex B (sasSubscribe.xsd).

13.3.3 RenewSubscription response example

A RenewSubscription operation response for SAS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<RenewSubscriptionResponse xmlns="http://www.opengis.net/sas" xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengeospatial.net/ows" xmlns:sos="http://www.opengis.net/sos"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd">
    <SubscriptionID>sub-1</SubscriptionID>
    <SubscriptionExpiration>
        <gml:TimeInstant>
            <gml:timePosition>2006-01-03T08:30:00</gml:timePosition>
        </gml:TimeInstant>
    </SubscriptionExpiration>
</RenewSubscriptionResponse>
```

13.3.4 RenewSubscription exceptions

When a SAS server encounters an error while performing a RenewSubscription operation, it shall return an exception report message as specified in Subclause 8 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 20. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 20.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 20 — Exception codes for RenewSubscription operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

14 CancelSubscription operation (mandatory)

14.1 Introduction

The CancelSubscription operation allows SAS clients to cancel a subscription.

14.2 CancelSubscription operation request

14.2.1 CancelSubscription request parameters

A request to perform the CancelSubscription operation shall include the parameters listed and defined in Table 21. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the "Name" column appear to contain spaces, they shall not contain spaces.

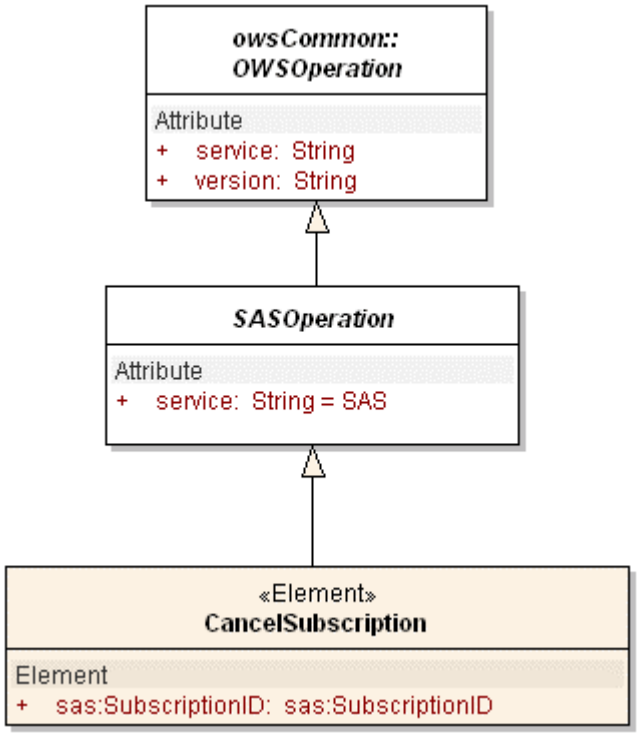


Figure 26: CancelSubscription in UML notation

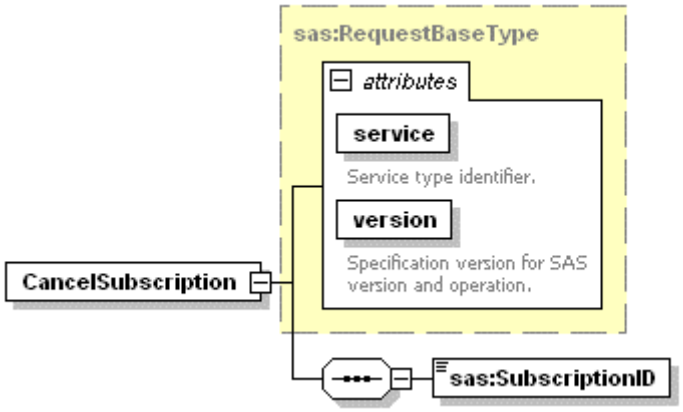


Figure 27: CancelSubscription in XMLSpy notation

Table 21 — Parameters in CancelSubscription operation request

Name a	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name (e.g., "GetCapabilities")	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
Subscription ID	Unique Identifier for the subscription. Provided by SAS server.	ID	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

The "Multiplicity and use" column in Table 21 specifies the optionality of each listed parameter and data structure in the CancelSubscription operation request. Since all parameters and data structures are mandatory in the operation request, all parameters and data structures shall be implemented by all SAS clients, using a specified value(s). Similarly, all parameters and data structures shall be implemented by all SAS servers, checking that each request parameter is received with any specified value(s).

14.2.2 CancelSubscription request KVP encoding

KVP encoding not supported!

14.2.3 CancelSubscription request XML encoding (mandatory)

All SAS servers shall implement HTTP POST transfer of the CancelSubscription operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a CancelSubscription operation request encoded in XML:

See annex B (sasSubscription.xsd).

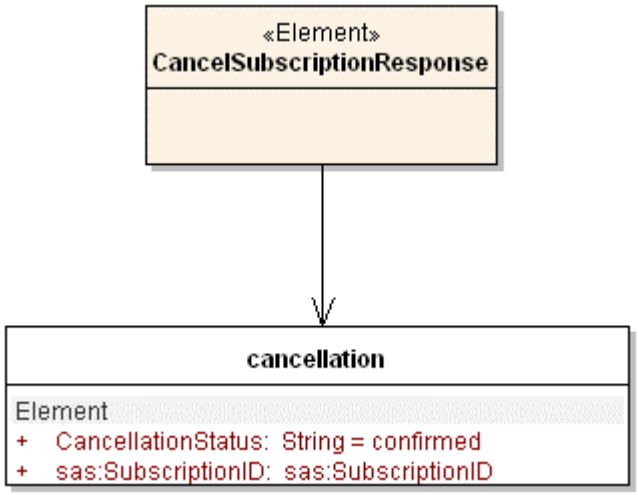
EXAMPLE An example CancelSubscription operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<CancelSubscription service="SAS" version="0.0.1" xmlns="http://www.opengis.net/sas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd">
    <SubscriptionID>sub-1</SubscriptionID>
</CancelSubscription>
```

14.3 CancelSubscription operation response

14.3.1 Normal response parameters

The normal response to a valid CancelSubscription operation request shall be CancelSubscriptionResponse. More precisely, a response from the CancelSubscription operation shall include the parts listed in Table 22. This table also specifies the UML model data type plus the multiplicity and use of each listed part.



[AW25]

Figure 28: CancelSubscriptionResponse in UML notation

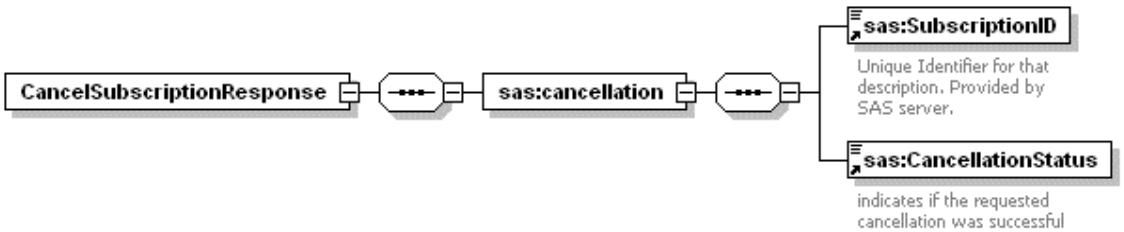


Figure 29: CancelSubscriptionResponse in XMLSpy notation

Table 22 — Parts of CancelSubscription operation response

Name	Definition	Data type and values	Multiplicity and use
cancellation	Element as defined below.		
cancellation.SubscriptionID	Unique Identifier for the subscription. Provided by SAS server.	ID	one (mandatory)
cancellation.CancellationStatus	Defines the status of the cancellation request	String “confirmed” or “already expired”	one (mandatory)

14.3.2 Normal response XML encoding

The following schema fragment specifies the contents and structure of a CancelSubscription operation response, always encoded in XML:

See annex B (sasSubscribe.xsd).

14.3.3 CancelSubscription response example

A CancelSubscription operation response for SAS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<CancelSubscriptionResponse xmlns="http://www.opengis.net/sas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sas
http://mars.uni-muenster.de/swerep\trunk\SAS\0.0.1\sasSubscribe.xsd">
    <cancellation>
        <SubscriptionID>sub-1</SubscriptionID>
        <CancellationStatus>confirmed</CancellationStatus>
    </cancellation>
</CancelSubscriptionResponse>
```

14.3.4 CancelSubscription exceptions

When a SAS server encounters an error while performing a CancelSubscription operation, it shall return an exception report message as specified in Subclause 8 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 23. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 23.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 23 — Exception codes for CancelSubscription operation

exceptionCode value	Meaning of code	“locator” value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported

MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter

Annex A
(normative)

Abstract test suite

A.1 General

TBD

In each Implementation Specification document, Annex A shall specify the Abstract Test Suite, as specified in Clause 9 and Annex A of ISO 19105. That Clause and Annex specify the ISO/TC 211 requirements for Abstract Test Suites. Examples of Abstract Test Suites are available in an annex of most ISO 191XX documents, one of the more useful is in ISO 191TBD. Note that this guidance may be more abstract than needed in an OpenGIS® Implementation Specification.

Annex B
(normative)

XML Schema Documents

In addition to this document, this specification includes several normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. After OGC acceptance of a Version 1.0.0 of this specification, these XML Schema Documents will also be posted online at the URL http://schemas.opengeospatial.net/SAS/0.30.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative.

These XML Schema Documents use and build on the OWS common XML Schema Documents specified [OGC 05-008], named:

ows19115subset.xsd

owsCommon.xsd

owsDataIdentification.xsd

owsExceptionReport.xsd

owsGetCapabilities.xsd

owsOperationsMetadata.xsd

owsServiceIdentification.xsd

owsServiceProvider.xsd

All these XML Schema Documents contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 05-008].

sasCommon.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sas="http://www.opengis.net/sas"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:gml="http://www.opengis.net/gml"
xmlns:swe="http://www.opengis.net/swe"
xmlns:om="http://www.opengis.net/om"
xmlns:ogc="http://www.opengis.net/ogc"
targetNamespace="http://www.opengis.net/sas"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.0.1" xml:lang="en">
    <annotation>
        <appinfo>sasCommon.xsd</appinfo>
        <documentation>
            <description>This XML Schema encodes the elements and types
that are shared by muliple SAS operations.</description>
        </documentation>
    </annotation>
    <!-- ================================================================
    and imports
    ================================================================ -->
    <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="./ows4sas.xsd"/>
    <import namespace="http://www.opengis.net/swe"
schemaLocation="../../sweCommon/1.0.30/sweCommon.xsd"/>
    <import namespace="http://www.opengis.net/om"
schemaLocation="../../om/1.0.30/observation.xsd"/>
    <import namespace="http://www.opengis.net/ogc"
schemaLocation="../../filter/1.1.30/filter.xsd"/>
    <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing/"/>
    <!-- ================================================================
    and types
    ================================================================ -->
    <complexType name="RequestBaseType">
        <annotation>
            <documentation>XML encoded SAS operation request base, for
all operations except Get Capabilities. In this XML encoding, no
"request" parameter is included, since the element name specifies the
specific operation. </documentation>
        </annotation>
        <attribute name="service" type="string" use="required"
fixed="SAS">
            <annotation>
                <documentation>Service type identifier. </documentation>
            </annotation>
        </attribute>
        <attribute name="version" type="string" use="required"
fixed="0.0.1">
            <annotation>
                <documentation>Specification version for SAS version and
operation.</documentation>
```

```
            </annotation>
         </attribute>
      </complexType>
      <element name="SubscriptionEndpoint"
type="wsa:EndpointReferenceType">
         <annotation>
            <documentation>A service endpoint reference that is intended
to be accessed by the requestor to receive the notification messages as
a result of this subscription.  Typically this will be a URL.  If the
protocol being used is XMPP the url will be of the form xmpp://jid@host
according to IETF draft draft-saintandre-xmpp-iri-01.  This allows a
client to connect to a server to receive notifications rather than to
require the client to accept connections from the server for posting
notifications.  This should be returned when the client has not sent a
ClientEndpoint reference to the server.</documentation>
         </annotation>
      </element>
      <element name="PublicationEndpoint"
type="wsa:EndpointReferenceType">
         <annotation>
            <documentation>A service endpoint reference that is intended
to be accessed by the pubisher to publish alerts.  Typically this will
be a URL.  If the protocol being used is XMPP the url will be of the
form xmpp://jid@host according to IETF draft draft-saintandre-xmpp-iri-
01.  This allows a client to connect to a server to publish
notifications rather than to require the client to accept connections
from the server.  This should be returned when the client has not sent
a ClientEndpoint reference to the server.</documentation>
         </annotation>
      </element>
      <element name="SystemDescriptionEndpoint"
type="wsa:EndpointReferenceType">
         <annotation>
            <documentation>A service endpoint reference that provides
access to a system description document</documentation>
         </annotation>
      </element>
      <element name="SubscribeEndpoint" type="wsa:EndpointReferenceType">
         <annotation>
            <documentation>A service endpoint reference for a Subscribe
operation.  This is used for supporting demand-based publishing as per
the OASIS WS-N model.  If the publisher passes this reference then the
publisher is not intending to publish alerts to the SAS. Instead, the
SAS should subscribe to the publisher when there are one or more
subscribers that are subscribing to an offering including the sensor
system(s) represented by this publisher.  If there is no active
subscription from the SAS then the publisher will not generate
alerts.</documentation>
         </annotation>
      </element>
      <element name="Result" type="anyType" nillable="true">
         <annotation>
            <documentation>This element is a container to hold the value
of some Observation in O&amp;amp;M since that schema does not have such an
element.  It is used to construct filter expressions that represent
alert conditions.</documentation>
         </annotation>
      </element>
```

```
    <element name="TransportBinding" type="anyURI">
       <annotation>
          <documentation>The URI represents the definition for the
given transport.  These are expeced to be defined in a GML dictionary.
Sampel values are
                     :ogc:def:transportbinding:xmpp:muc and
urn:ogc:def:transportbinding:http:post
                              </documentation>
       </annotation>
    </element>
    <element name="PublicationID" type="ID">
       <annotation>
          <documentation>ID administered by SAS server to uniquely
identify advertisement offerings.</documentation>
       </annotation>
    </element>
    <element name="DesiredPublicationExpiration">
       <annotation>
          <documentation>The time that the requestor would like the
publication channel  to expire.  The server may grant a shorter
validity time than requested.</documentation>
       </annotation>
       <complexType>
          <sequence>
             <element ref="gml:TimeInstant"/>
          </sequence>
       </complexType>
    </element>
    <element name="PublicationExpiration">
       <annotation>
          <documentation>The time that this publicaiton channel
automatically terminates.</documentation>
       </annotation>
       <complexType>
          <sequence>
             <element ref="gml:TimeInstant"/>
          </sequence>
       </complexType>
    </element>
    <element name="SubscriptionOfferingID" type="token">
       <annotation>
          <documentation>This unique identifier identifies the
SubscriptionOffering. It is part of the Capabilities document and can
be used by clients in Subscribe operation requests.</documentation>
       </annotation>
    </element>
    <element name="AlertTime">
       <annotation>
          <documentation>Specifies time periods for which alerts are
avialable. </documentation>
       </annotation>
       <complexType>
          <sequence>
             <element ref="ogc:temporalOps"/>
          </sequence>
       </complexType>
    </element>
    <element name="Procedure" type="om:ProcedurePropertyType">
```

```
      <annotation>
         <documentation>Reference to a particular sensor system
advertised in the offering. Allows client to request alerts from one or
more sensors in the offering. This is to support scenarios with sensor
grids (we don't want to have one offering for each sensor in that
case). Note that sensorML can describe Sensor Arrays too.
</documentation>
      </annotation>
   </element>
   <element name="ObservedProperty" type="anyURI">
      <annotation>
         <documentation>ID of a phenomenon advertised in capabilities
document.
                        possible phenomena are listed in the selected
offering capabilities.
                        </documentation>
      </annotation>
   </element>
   <element name="ResultDefinition">
      <annotation>
         <documentation>Alerts are reported only when the result of an
associated observation match this expression.</documentation>
      </annotation>
      <complexType>
         <choice>
            <element ref="ogc:comparisonOps"/>
            <element ref="ogc:logicOps"/>
         </choice>
      </complexType>
   </element>
   <element name="ResultFormat" type="ows:MimeType">
      <annotation>
         <documentation>Mime type of the reponse</documentation>
      </annotation>
   </element>
   <element name="CancellationStatus">
      <annotation>
         <documentation>indicates if the requested cancellation was
successful</documentation>
      </annotation>
      <simpleType>
         <restriction base="string">
            <enumeration value="confirmed"/>
            <enumeration value="already expired"/>
         </restriction>
      </simpleType>
   </element>
</schema>
```

sasAdvertise.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp1 U (http://www.altova.com) by Ingo
Simonis (Institute for Geoinformatics) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:sas="http://www.opengis.net/sas"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ns1="http://www.opengeospatial.net/sos"
targetNamespace="http://www.opengis.net/sas"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.0.31" xml:lang="en">
   <annotation>
      <appinfo>sasPublish.xsd</appinfo>
      <documentation>
         <dc:description>This XML Schema defines the SAS Publish
request XML elements and types.</dc:description>
      </documentation>
   </annotation>
   <!-- ============================================================
   and imports
   ============================================================ -->
   <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../ows/1.0.30/owsCommon.xsd"/>
   <import namespace="http://www.opengis.net/ogc"
schemaLocation="../../filter/1.1.30/filter.xsd"/>
   <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/temporal.xsd"
/>
   <import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing/"/>
   <include schemaLocation="sasCommon.xsd"/>
   <!-- ============================================================

   ============================================================ -->
   <element name="Advertise">
      <annotation>
         <documentation>Request to a SAS to allow a publisher to
publish alerts.</documentation>
      </annotation>
      <complexType>
         <complexContent>
            <extension base="sas:RequestBaseType">
               <sequence>
                  <element name="AdvertisementID" type="anyURI"
minOccurs="0">
                     <annotation>
                        <documentation>ID of an offering that might
be advertised in the capabilities document as it is if unambiguousness
has been insured.
                        following parameters are depending on the
selected offering.
                        </documentation>
```

```
                        </annotation>
                    </element>
                    <element name="AlertTime" minOccurs="0"
maxOccurs="unbounded">
                        <annotation>
                            <documentation>Alert time periods that this
publisher will publish.  This adds a way to specify the publication of
historical alerts.
                            </documentation>
                        </annotation>
                        <complexType>
                            <sequence>
                                <element ref="ogc:temporalOps"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element ref="sas:SystemDescriptionEndpoint"
maxOccurs="unbounded">
                        <annotation>
                            <documentation>References to ervice endpoints
that can return sensor system descriptions for systems for which the
publisher is publishing alerts</documentation>
                        </annotation>
                    </element>
                    <element name="ObservedProperty" type="anyURI"
minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <documentation>Observables that are
associated with the alerts.</documentation>
                        </annotation>
                    </element>
                    <element name="FeatureOfInterest" minOccurs="0">
                        <annotation>
                            <documentation>Specifies target feature for
which alerts are published. Mostly a helper for in-situ sensors, since
geo-location has to be done on the server side. The supported area
should be listed in the selected offering capabilities.
                            </documentation>
                        </annotation>
                        <complexType>
                            <choice>
                                <element ref="ogc:spatialOps"/>
                                <element name="ObjectID" type="anyURI"
maxOccurs="unbounded">
                                    <annotation>
                                        <documentation>Unordered list of
zero or more object identifiers. These identifiers are usually listed
in the Contents section of the service metadata (Capabilities)
document. If no ObjectID value is included, and if only one category of
objects is allowed for this operation, the server shall return all
objects of that category. NOTE: Although retention of this ability is
allowed by a specific OWS that uses this operation, such retention is
discouraged due to possible problems. Making this ability optional
implementation by servers reduces interoperability. Requiring
implementation of this ability can require a server to return a huge
response, when there are a large number of items in that category.
</documentation>
                                    </annotation>
```

```
                        </element>
                     </choice>
                  </complexType>
               </element>
               <element name="Result">
                  <annotation>
                     <documentation>The publihser reports alerts
where the result of an associated observation matches this expression.
                     </documentation>
                  </annotation>
                  <complexType>
                     <choice>
                        <element ref="ogc:comparisonOps"/>
                        <element ref="ogc:logicOps"/>
                     </choice>
                  </complexType>
               </element>
               <element ref="sas:DesiredPublicationExpiration"
minOccurs="0"/>
               <element name="AlertFormat" type="ows:MimeType">
                  <annotation>
                     <documentation>ID of the format to be used
for the alert data.</documentation>
                  </annotation>
               </element>
               <element ref="sas:TransportBinding"/>
               <element ref="sas:SubscribeEndpoint"
minOccurs="0"/>
            </sequence>
            <attribute ref="gml:id"/>
         </extension>
      </complexContent>
   </complexType>
</element>
<element name="AdvertiseResponse">
   <complexType>
      <sequence>
         <element ref="sas:PublicationID"/>
         <element ref="sas:PublicationExpiration" minOccurs="0"/>
         <element ref="sas:PublicationEndpoint"/>
         <element ref="sas:SystemDescriptionEndpoint"
minOccurs="0">
            <annotation>
               <documentation>Reference to a service endpoint on
the SAS server that refers to the sensor system that the publisher
included in the request.  The purpose of this is to allow the SAS to
communicate the sesnor regisrtration that it has added in reponse to
seeing a new sensor.</documentation>
            </annotation>
         </element>
      </sequence>
      <attribute ref="gml:id"/>
   </complexType>
</element>
<element name="CancelAdvertisement">
   <complexType>
      <complexContent>
         <extension base="sas:RequestBaseType">
```

```
                <sequence>
                    <element ref="sas:PublicationID"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</element>
<element name="CancelAdvertisementResponse">
    <complexType>
        <sequence>
            <element name="cancellation" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element ref="sas:PublicationID"/>
                        <element ref="sas:CancellationStatus"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
</element>
<element name="RenewAdvertisement">
    <complexType>
        <complexContent>
            <extension base="sas:RequestBaseType">
                <sequence>
                    <element ref="sas:PublicationID"/>
                    <element ref="sas:DesiredPublicationExpiration"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</element>
<element name="RenewAdvertisementResponse">
    <complexType>
        <sequence>
            <element ref="sas:PublicationID"/>
            <element ref="sas:PublicationExpiration" minOccurs="0"/>
            <element name="renewalStatus">
                <annotation>
                    <documentation>Indicator shows if renewal request
was confirmed or rejected.</documentation>
                </annotation>
                <simpleType>
                    <restriction base="string">
                        <enumeration value="confirmed"/>
                        <enumeration value="rejected"/>
                    </restriction>
                </simpleType>
            </element>
        </sequence>
    </complexType>
</element>
</schema>
```

sasSubscribe.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp1 U (http://www.altova.com) by Ingo
Simonis (Institute for Geoinformatics) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:sas="http://www.opengis.net/sas"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:sos="http://www.opengis.net/sos"
xmlns:swe="http://www.opengis.net/swe"
targetNamespace="http://www.opengis.net/sas"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.0.31" xml:lang="en">
   <annotation>
      <appinfo>sasSubscribe.xsd</appinfo>
      <documentation>
         <dc:description>This XML Schema defines the SAS Subscribe
request XML elements and types.</dc:description>
      </documentation>
   </annotation>
   <!-- ============================================================
   and imports
   ============================================================ -->
   <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../ows/1.0.30/owsCommon.xsd"/>
   <import namespace="http://www.opengis.net/ogc"
schemaLocation="../../filter/1.1.30/filter.xsd"/>
   <import namespace="http://www.opengis.net/swe"
schemaLocation="../../sweCommon/1.0.30/sweCommon.xsd"/>
   <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/temporal.xsd"
/>
   <import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing/"/>
   <include schemaLocation="sasCommon.xsd"/>
   <!-- ============================================================

   ============================================================ -->
   <element name="Subscribe">
      <annotation>
         <documentation>Request to a SAS to subscribe to
alerts.</documentation>
      </annotation>
      <complexType>
         <complexContent>
            <extension base="sas:RequestBaseType">
               <sequence>
                  <element ref="sas:SubscriptionOfferingID"/>
                  <element ref="sas:AlertTime" minOccurs="0"
maxOccurs="unbounded"/>
                  <element ref="sas:Procedure" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                    <element ref="sas:ObservedProperty" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="swe:featureOfInterest"
minOccurs="0"/>
                    <element ref="sas:ResultDefinition" minOccurs="0"/>
                    <element name="DesiredSubscriptionExpiration"
minOccurs="0">
                       <annotation>
                          <documentation>The time that the requestor
would like the subscription to expire.  The server may grant a shorter
validity time than requested.</documentation>
                       </annotation>
                       <complexType>
                          <sequence>
                             <element ref="gml:TimeInstant"/>
                          </sequence>
                       </complexType>
                    </element>
                    <element ref="sas:ResultFormat"/>
                    <element ref="sas:TransportBinding"/>
                 </sequence>
                 <attribute ref="gml:id"/>
              </extension>
           </complexContent>
        </complexType>
     </element>
     <element name="SubscribeResponse">
        <complexType>
           <sequence>
              <element ref="sas:SubscriptionID"/>
              <element name="SubscriptionExpiration">
                 <annotation>
                    <documentation>The time that this subscription
automatically terminates.</documentation>
                 </annotation>
                 <complexType>
                    <sequence>
                       <element ref="gml:TimeInstant"/>
                    </sequence>
                 </complexType>
              </element>
              <element ref="sas:ResultFormat"/>
              <element ref="sas:SubscriptionEndpoint"/>
              <element ref="sas:SystemDescriptionEndpoint"
minOccurs="0"/>
           </sequence>
           <attribute ref="gml:id"/>
        </complexType>
     </element>
     <element name="CancelSubscription">
        <complexType>
           <complexContent>
              <extension base="sas:RequestBaseType">
                 <sequence>
                    <element name="SubscriptionID" type="ID"/>
                 </sequence>
              </extension>
           </complexContent>
```

```
      </complexType>
   </element>
   <element name="CancelSubscriptionResponse">
      <complexType>
         <sequence>
            <element name="cancellation">
               <complexType>
                  <sequence>
                     <element ref="sas:SubscriptionID"/>
                     <element ref="sas:CancellationStatus"/>
                  </sequence>
               </complexType>
            </element>
         </sequence>
      </complexType>
   </element>
   <element name="RenewSubscription">
      <complexType>
         <complexContent>
            <extension base="sas:RequestBaseType">
               <sequence>
                  <element name="SubscriptionID" type="ID"/>
                  <element name="DesiredSubscriptionExpiration">
                     <annotation>
                        <documentation>The time that the requestor
would like the subscription to expire.  The server may grant a shorter
validity time than requested.</documentation>
                     </annotation>
                     <complexType>
                        <sequence>
                           <element ref="gml:TimeInstant"/>
                        </sequence>
                     </complexType>
                  </element>
               </sequence>
            </extension>
         </complexContent>
      </complexType>
   </element>
   <element name="RenewSubscriptionResponse">
      <complexType>
         <sequence>
            <element ref="sas:SubscriptionID"/>
            <element ref="sas:SubscriptionExpiration"/>
         </sequence>
      </complexType>
   </element>
   <!---->
   <element name="SubscriptionExpiration">
      <annotation>
         <documentation>The time that this subscription automatically
terminates.</documentation>
      </annotation>
      <complexType>
         <sequence>
            <element ref="gml:TimeInstant"/>
         </sequence>
      </complexType>
```

```
    </element>
    <element name="SubscriptionID" type="ID">
        <annotation>
            <documentation>Unique Identifier for that description.
Provided by SAS server.</documentation>
        </annotation>
    </element>
</schema>
```

sasContents.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:sas="http://www.opengis.net/sas"
xmlns:gml="http://www.opengis.net/gml"
xmlns:om="http://www.opengis.net/om"
xmlns:swe="http://www.opengis.net/swe"
xmlns:ogc="http://www.opengis.net/ogc"
targetNamespace="http://www.opengis.net/sas"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.0.1" xml:lang="en">
   <annotation>
      <appinfo>sasContents.xsd</appinfo>
      <documentation>
         <description>This XML Schema encodes the Contents section of
the SAS GetCapabilities operation response.</description>
      </documentation>
   </annotation>
   <!-- ================================================================
   and imports
   ================================================================ -->
   <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
   <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="./ows4sas.xsd"/>
   <import namespace="http://www.opengis.net/om"
schemaLocation="../../om/1.0.30/observation.xsd"/>
   <import namespace="http://www.opengis.net/swe"
schemaLocation="../../sweCommon/1.0.30/recordSchema.xsd"/>
   <import namespace="http://www.opengis.net/ogc"
schemaLocation="../../filter/1.1.30/filter.xsd"/>
   <include schemaLocation="sasCommon.xsd"/>
   <!-- ================================================================
   and types
   ================================================================ -->
   <element name="Contents">
      <annotation>
         <documentation/>
      </annotation>
      <complexType>
         <sequence>
            <element name="SubscriptionOfferingList" minOccurs="0">
               <annotation>
                  <documentation>root element for all
SubscriptionOfferings</documentation>
               </annotation>
               <complexType>
                  <sequence>
                     <element name="SubscriptionOffering"
type="sas:SubscriptionOfferingType" maxOccurs="unbounded"/>
                  </sequence>
               </complexType>
            </element>
            <element name="AdvertisementOfferingList" minOccurs="0">
```

```
                <annotation>
                    <documentation>root element for all
AdvertisementOfferings</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="AdvertisementOffering"
type="sas:PublishAlertOfferingType" maxOccurs="unbounded"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
</element>
<complexType name="AlertOfferingBaseType" abstract="true">
    <complexContent>
        <restriction base="gml:AbstractFeatureType">
            <sequence>
                <group ref="gml:StandardObjectProperties"/>
                <element ref="gml:boundedBy" minOccurs="0"/>
            </sequence>
        </restriction>
    </complexContent>
</complexType>
<complexType name="SubscriptionOfferingType">
    <complexContent>
        <extension base="sas:AlertOfferingBaseType">
            <sequence>
                <element ref="sas:SubscriptionOfferingID"/>
                <element name="AlertTime" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="Procedure"
type="om:ProcedurePropertyType" maxOccurs="unbounded"/>
                <element name="ObservedProperty"
type="swe:PhenomenonPropertyType" maxOccurs="unbounded"/>
                <element name="FeatureOfInterest"
type="gml:FeaturePropertyType"/>
                <element ref="sas:ResultDefinition" minOccurs="0"/>
                <element ref="sas:ResultFormat" maxOccurs="unbounded"/>
                <element ref="sas:TransportBinding"
maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="PublishAlertOfferingType">
    <complexContent>
        <extension base="sas:AlertOfferingBaseType">
            <sequence>
                <element name="ObservedProperty"
type="swe:PhenomenonPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <element name="AlertFormat" type="ows:MimeType"
maxOccurs="unbounded"/>
                <element ref="sas:TransportBinding"
maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
```

```
    </complexType>
    <element name="GeoReferenceableFeature"
type="gml:AbstractFeatureType" substitutionGroup="gml:_Feature"/>
</schema>
```

sasGetCapabilities.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:sas="http://www.opengis.net/sas"
targetNamespace="http://www.opengis.net/sas"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.0.1" xml:lang="en">
   <annotation>
      <appinfo>sosGetCapabilities.xsd 2005/05/11</appinfo>
      <documentation>
         <description>This XML Schema encodes the SAS GetCapabilities
operation request and response.</description>
      </documentation>
   </annotation>
   <!-- ================================================================
   and imports
   ================================================================ -->
   <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="./ows4sas.xsd"/>
   <import namespace="http://www.opengis.net/ogc"
schemaLocation="./ogc4sas.xsd"/>
   <include schemaLocation="sasContents.xsd"/>
   <!-- ================================================================
   and types
   ================================================================ -->
   <element name="GetCapabilities">
      <annotation>
         <documentation>Request to a SAS to perform the
GetCapabilities operation. This operation allows a client to retrieve
service metadata (capabilities XML) providing metadata for the specific
SAS server. In this XML encoding, no "request" parameter is included,
since the element name specifies the specific operation.
</documentation>
      </annotation>
      <complexType>
         <complexContent>
            <extension base="ows:GetCapabilitiesType">
               <attribute name="service" type="ows:ServiceType"
use="required" fixed="SAS"/>
            </extension>
         </complexContent>
      </complexType>
   </element>
   <!-- ================================================================ -->
   <element name="Capabilities">
      <annotation>
         <documentation>XML encoded SAS GetCapabilities operation
response. This document provides clients with service metadata about a
specific service instance, including metadata about the tightly-coupled
data served. If the server does not implement the updateSequence
parameter, the server shall always return the complete Capabilities
document, without the updateSequence parameter. When the server
implements the updateSequence parameter and the GetCapabilities
operation request included the updateSequence parameter with the
current value, the server shall return this element with only the
```

```
"version" and "updateSequence" attributes. Otherwise, all optional
elements shall be included or not depending on the actual value of the
Sections parameter in the GetCapabilities operation request.
</documentation>
       </annotation>
       <complexType>
          <complexContent>
             <extension base="ows:CapabilitiesBaseType">
                <sequence>
                   <element ref="ogc:Filter_Capabilities"
minOccurs="0"/>
                   <element ref="sas:Contents"/>
                </sequence>
             </extension>
          </complexContent>
       </complexType>
   </element>
</schema>
```

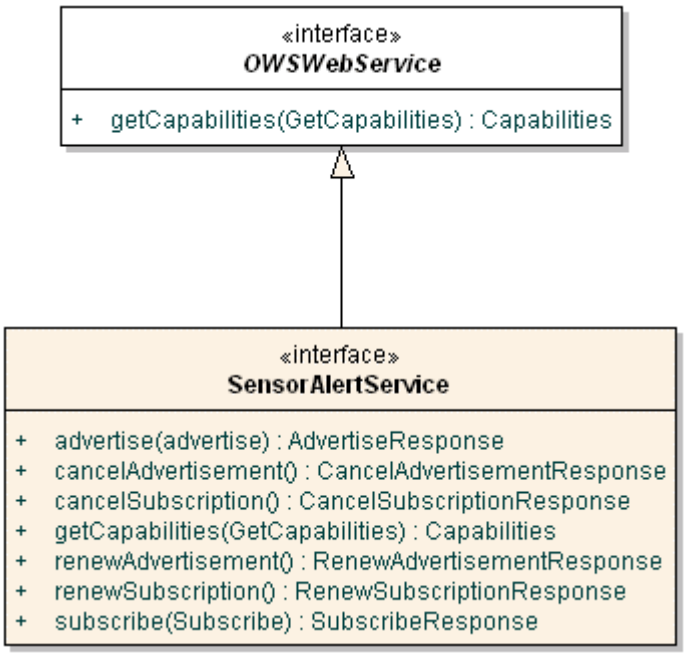
Annex C
(informative)

UML model

C.1 Introduction

This annex provides a UML model of the SAS interface, using the OGC/ISO profile of UML summarized in Subclause 5.3 of [05-008].

Figure C.1 is a simple UML diagram summarizing the SAS interface. This class diagram shows that the SAS class inherits the getCapabilities operation from the OGCWebService interface class, and adds the SAS operations. (The capitalization of names uses the OGC/ISO profile of UML.)



[AW26]

Figure C.1 — SAS interface UML diagram

Each of the SAS operations uses a request and a response data type, each of which is defined by one or more additional UML classes. The following subclauses provide a more complete UML model of the SAS interface, adding UML classes defining the operation request and response data types.

Annex D
(informative)

Example XML documents

D.1 Introduction

This annex provides more example XML documents than given in the body of this document.

D.2 GetCapabilities response

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/sas" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengeospatial.net/ows" xmlns:om="http://www.opengis.net/om"
xmlns:gml="http://www.opengis.net/gml" xmlns:swe="http://www.opengis.net/swe"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/sas
http://mars.uni-muenster.de/swerep/trunk/sas/0.0.1/sasGetCapabilities.xsd" version="0.0.1">
    <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <!-- Service Identification -->
    <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <ows:ServiceIdentification>
        <ows:Title>ifgi Sensor Alert Service</ows:Title>
        <ows:Abstract/>
        <ows:Keywords>
            <ows:Keyword>gage height, water level</ows:Keyword>
        </ows:Keywords>
        <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:SAS</ows:ServiceType>
        <ows:ServiceTypeVersion>0.0.1</ows:ServiceTypeVersion>
        <ows:Fees>NONE</ows:Fees>
        <ows:AccessConstraints>NONE</ows:AccessConstraints>
    </ows:ServiceIdentification>
    <!--~~~~~~~~~~~~~~~~~~~~~~~~-->
    <!-- Provider Description -->
    <!--~~~~~~~~~~~~~~~~~~~~~~~~-->
    <ows:ServiceProvider>
        <ows:ProviderName>Institue for Geoinformatics, University of Muenster</ows:ProviderName>
        <ows:ProviderSite xlink:href="http://ifgi.uni-muenster.de"/>
        <ows:ServiceContact>
            <ows:IndividualName>Alexander C. Walkowski</ows:IndividualName>
            <ows:PositionName>Research Associate</ows:PositionName>
            <ows:ContactInfo>
                <ows:Phone>
                    <ows:Voice>+49-251-83-30056</ows:Voice>
                </ows:Phone>
                <ows:Address>
                    <ows:DeliveryPoint>Rober-Koch-Str 26-28</ows:DeliveryPoint>
                    <ows:City>Muenster</ows:City>
                    <ows:AdministrativeArea>NRW</ows:AdministrativeArea>
                    <ows:PostalCode>48149</ows:PostalCode>
                    <ows:Country>Germany</ows:Country>
                    <ows:ElectronicMailAddress>walkowski@uni-muenster.de</ows:ElectronicMailAddress>
                </ows:Address>
```

```
            </ows:ContactInfo>
        </ows:ServiceContact>
    </ows:ServiceProvider>
    <!--********************************-->
    <!-- Operations Metadata -->
    <!--********************************-->
    <ows:OperationsMetadata>
        <ows:Operation name="GetCapabilities">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://mars.uni-muenster.de:8080/SAS/sas?"/>
                    <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="service" use="required">
                <ows:Value>SAS</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="AcceptVersions" use="optional">
                <ows:Value>0.0.1</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="Sections" use="optional">
                <ows:Value>All</ows:Value>
                <ows:Value>Contents</ows:Value>
                <ows:Value>ServiceIdentification</ows:Value>
                <ows:Value>ServiceProvider</ows:Value>
                <ows:Value>OperationsMetadata</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="updateSequence" use="optional">
                <ows:Value>jan022006</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="AcceptFormats" use="optional">
                <ows:Value>text/xml</ows:Value>
            </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="Advertise">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="AdvertisementID" use="optional">
                <ows:Value>pub-1</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="AlertTime" use="optional">
                <ows:Range/>
            </ows:Parameter>
            <ows:Parameter name="SystemDescriptionEndpoint" use="required">
                <ows:Value>http://mars.uni-muenster.de:8080/OWS3-SOS/sos</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="ObservedProperty" use="optional">
                <ows:Value>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="FeatureOfInterest" use="optional">
                <ows:Value>na</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="Result" use="required">
                <ows:Value/>
            </ows:Parameter>
            <ows:Parameter name="DesiredPublicationExpiration" use="optional">
                <ows:Value/>
            </ows:Parameter>
            <ows:Parameter name="AlertFormat" use="required">
                <ows:Value>text/xml;subtype&amp;quot;tml/2.0&amp;quot;</ows:Value>
```

```
        </ows:Parameter>
        <ows:Parameter name="TransportBinding" use="required">
            <ows:Value>urn:ogc:def:transportbinding:xmpp:muc</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="SubscribeEndpoint" use="optional">
            <ows:Value>xmpp://sub-1@mars.uni-muenster.de</ows:Value>
        </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="CancelAdvertisement">
        <ows:DCP>
            <ows:HTTP>
                <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
            </ows:HTTP>
        </ows:DCP>
        <ows:Parameter name="PublicationID" use="required">
            <ows:Value>pub-1</ows:Value>
        </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="RenewAdvertisement">
        <ows:DCP>
            <ows:HTTP>
                <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
            </ows:HTTP>
        </ows:DCP>
        <ows:Parameter name="PublicationID" use="required">
            <ows:Value>pub-1</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="DesiredPublicationExpiration" use="required">
            <ows:Value>na</ows:Value>
        </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="Subscribe">
        <ows:DCP>
            <ows:HTTP>
                <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
            </ows:HTTP>
        </ows:DCP>
        <ows:Parameter name="SubscriptionOfferingID">
            <ows:Value>subOf-1</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="AlertTime">
            <ows:Range/>
        </ows:Parameter>
        <ows:Parameter name="ObservedProperty" use="optional">
            <ows:Value>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="Procedure" use="optional">
            <ows:Value>system.ifgi-sensor-1</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="FeatureOfInterest" use="optional">
            <ows:Value>na</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="ResultDefinition" use="optional">
            <ows:Value>na</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="DesiredSubscriptionExpiration" use="optional">
            <ows:Value>na</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="ResultFormat" use="required">
            <ows:Value>txt/xml;subtype=&quot;tml/2.0&quot;</ows:Value>
        </ows:Parameter>
        <ows:Parameter name="TransportBinding" use="required">
            <ows:Value>urn:ogc:def:transportbinding:xmpp:muc</ows:Value>
```

```
            </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="CancelSubscription">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="SubscriptionID" use="required">
                <ows:Value>sub-1</ows:Value>
            </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="RenewSubscription">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Post xlink:href="http://mars.uni-muenster.de:8080/SAS/sas"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="SubscriptionID" use="required">
                <ows:Value>sub-1</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="DesiredSubscriptionExpiration" use="required">
                <ows:Value>na</ows:Value>
            </ows:Parameter>
        </ows:Operation>
    </ows:OperationsMetadata>
    <!--================================================================-->
    <!--FILTER CAPABILITIES SECTION-->
    <!--================================================================-->
    <ogc:Filter_Capabilities>
        <ogc:Spatial_Capabilities>
            <ogc:GeometryOperands>
                <ogc:GeometryOperand>gml:Envelope</ogc:GeometryOperand>
                <!-- This operand is not supported, but the FilterCapabilities spec lacks the ability of
ogc:Spatial_Capabilities=nil!-->
            </ogc:GeometryOperands>
            <ogc:SpatialOperators>
                <ogc:SpatialOperator name="BBOX"></ogc:SpatialOperator>
                <!-- This operand is not supported, but the FilterCapabilities spec lacks the ability of
ogc:SpatialOperators=nil!-->
            </ogc:SpatialOperators>
        </ogc:Spatial_Capabilities>
        <ogc:Scalar_Capabilities>
            <ogc:ComparisonOperators>
                <ogc:ComparisonOperator>GreaterThan</ogc:ComparisonOperator>
                <ogc:ComparisonOperator>LessThan</ogc:ComparisonOperator>
            </ogc:ComparisonOperators>
        </ogc:Scalar_Capabilities>
        <ogc:Id_Capabilities>
            <ogc:EID></ogc:EID>
        </ogc:Id_Capabilities>
        <ogc:Temporal_Capabilities>
            <ogc:TemporalOperators>
                <ogc:TemporalOperator></ogc:TemporalOperator>
            </ogc:TemporalOperators>
        </ogc:Temporal_Capabilities>
    </ogc:Filter_Capabilities>
    <!--**********************************-->
    <!-- Contents Section -->
    <!--**********************************-->
    <Contents>
        <SubscriptionOfferingList>
            <SubscriptionOffering>
```

```
                <gml:boundedBy>
                    <gml:Envelope srsName="EPSG:4326">
                        <gml:lowerCorner>7.727957 51.883905</gml:lowerCorner>
                        <gml:upperCorner>7.727959 51.883907</gml:upperCorner>
                    </gml:Envelope>
                </gml:boundedBy>
                <SubscriptionOfferingID>subOf-1</SubscriptionOfferingID>
                <Procedure xlink:href="system.ifgi-sensor-1"/>
                <:ObservedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel"/>
                <FeatureOfInterest>
                    <om:Station>
                        <gml:name>ALBER</gml:name>
                        <om:position>
                            <gml:Point>
                                <gml:pos srsName="urn:ogc:crs:epsg4326">7.727958 51.883906</gml:pos>
                            </gml:Point>
                        </om:position>
                        <om:procedureHosted xlink:href="system.ifgi-sensor-1"/>
                    </om:Station>
                </FeatureOfInterest>
                <ResultDefinition>
                    <ogc:PropertyIsGreaterThan>
                        <ogc:Literal>130</ogc:Literal>
                    </ogc:PropertyIsGreaterThan>
                </ResultDefinition>
                <ResultFormat>text/xml;subtype=&quot;tml/2.0&quot;</ResultFormat>
                <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
            </SubscriptionOffering>
        </SubscriptionOfferingList>
        <AdvertisementOfferingList>
            <AdvertisementOffering gml:id="pub-1">
                <gml:boundedBy>
                    <gml:Envelope srsName="EPSG:4326">
                        <gml:lowerCorner>7.727957 51.883905</gml:lowerCorner>
                        <gml:upperCorner>7.727959 51.883907</gml:upperCorner>
                    </gml:Envelope>
                </gml:boundedBy>
                <ObservedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel"/>
                <AlertFormat>text/xml;subtype=&quot;tml/2.0&quot;</AlertFormat>
                <TransportBinding>urn:ogc:def:transportbinding:xmpp:muc</TransportBinding>
            </AdvertisementOffering>
        </AdvertisementOfferingList>
    </Contents>
</Capabilities>
```

Annex E
(informative)

Messaging Server: XMPP based implementation

C.1 Introduction

This annex provides a description of transport bindings and ResultFormats used in the Interoperability Experiment Sensor Alert Service (IE SAS) conducted during the summer of 2005. The annex does not provide an exhaustive listing of possible transport bindings and result formats.

C.2 XMPP-transport binding

The service metadata document contains in the Contents section both AdvertisementOfferings and SubscriptoinOfferings. The offerings provide information about the transport mechanism (transport binding) that will be used either for the dissemination of events to the consumer or the publication of events by the producer. Within the service metadata the transport binding is identified by an URI. It is expected that these transport bindings are defined in a GML dictionary. This annex defines the XMPP transport binding that will be represented by the following URI:

```
urn:ogc:def:transportbinding:xmpp:muc
```

Upon receiving an advertise request, the SAS will return the PublicationEndpoint in form of an URL. According to the IETF draft draft-saintandre-xmpp-iri-01 the URL will be of the form xmpp://jid@host. This allows an event producer to connect to the messaging server part (in this case the XMPP server) of the SAS to publish notifications rather than to require the producer to accept connections form the server. The URL identifies a XMPP Multi User Chat (MUC), to which the producer will send all notifications of the advertised type. The format of the notifications is defined by the AlertFormat parameter of the advertise operation.

The outgoing interface of the SAS is quite similar to the incoming one described above. The consumer will issue a subscribe request and will get back the SubscriptionEndpoint, which is equivalent to the PublicationEndpoint and encoded as URL. The SubscriptionEndpoint is intended to be accessed by the requestor to receive the notification messages as a result of this subscription. The URL identifies the XMPP MUC in which the consumer can receive all alerts that match his subscription. The use of the XMPP MUC protocol allows for an event based distribution of the alerts. The consumer does not need to request for alerts. Instead the messaging server part of the SAS (in this case the XMPP server) will deliver the alerts instantaneously. This allows a consumer to connect to a server to receive notifications rather than to require the client to

accept connections from the server for posting notifications. The format of the delivered notification can be defined by the ResultFormat parameter of the subscribe operation.

In the simplest case the Subscription Endpoint is equal to the PublicationEndpoint. This is the case if the consumer is interested in the same events advertised and published in the desired format by the producer. But most often the consumer may only be interested in a subset of the published events. Thus the subscribe operation provides a filter mechanism. A spatial filter may be defined by the featureOfInterest parameter, a temporal filter by the AlertTime parameter, and a thematic filter by means of the ResultDefinition parameter.

If the published events do not exactly match to the subscription of the consumer, the relationship between the PublicationEndpoint and the SubscriptionEndpoint is as follows:

The producer will publish alerts in the advertised format by sending them to the PublicationEndpoint. The SAS will screen all incoming alerts and disseminate them to the appropriate SubscriptionEndpoint (MUC). Upon receiving a subscribe request the SAS will check if an equivalent subscription already exist. If this check evaluates to true, the subscribe response is the SubscriptionEndpoint referencing to the existing MUC. In the case the subscribe request defines a new subscription, the SAS will create a new MUC and returns a SubscriptionEndpoint referencing to this MUC.

C.3 TML-alert format

Both advertise and subscribe operation allow the definition of the alert/result format. This clause describes the TML alert format used during the IE SAS. The TML-alert format consists of the TML <data> tag. The content of the <data> tag is defined as follows:

```
<data dateTime=... reference=...>LEVEL EXPIRATION VALUE LAT LON
ALT</data>
```

LEVEL	The level is a kind of quality identifier for the alert. For example level 1 might mean a raw alert and a level 2 might mean a processed alert. In other cases there might need to be a finer division of levels.
EXPIRATION	gives the time in milliseconds for the automatic expiration of the alert. A value of 0 (zero) indicates no expiration and a value of -1 can be used for alert cancellation.
VALUE	is the alert value; the value observed by the sensor.
LAT	is the latitude component of the sensors position producing the alert.
LON	is the longitude component of the sensors position producing the alert.
ALT	is the altitude component of the sensors position producing the alert.

The format will be identified by the following mime-type:

`text/xml;subtype="tml2.0"`[AW27]

C.3 CAP-alert format

CAP-alert format stands for the Common Alerting Protocol alert format. The payload of the XMPP message consists of a valid CAP message. The format will be identified by the following mime-type:

`text/xml;subtype="cap1.0"`

Bibliography

[1] Guidelines for Successful OGC Interface Specifications, OGC document 00-014r1