

# Open Geospatial Consortium

Date: 2010-10-12

Reference number of this document: OGC 09-112r1

Category: Public Discussion Paper

Editor(s): Simon Jirka, Arne Bröring, Daniel Nüst

## **OGC<sup>®</sup> Sensor Observable Registry (SOR) Discussion Paper**

Copyright © 2010 Open Geospatial Consortium.

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

### **Warning**

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

<b>Contents</b>		<b>Page</b>
1	Scope.....	1
2	Normative references .....	2
3	Terms and definitions .....	3
4	Conventions .....	4
4.1	Abbreviated terms .....	4
5	Sensor Observable Registry Overview .....	5
6	GetCapabilities operation (mandatory).....	6
6.1	Introduction .....	6
6.2	Operation request .....	6
6.3	GetCapabilities operation response .....	7
6.3.1	Exceptions.....	11
7	GetDefinitionURIs operation (mandatory).....	12
7.1	Introduction .....	12
7.2	GetDefinitionURIs operation request.....	12
7.2.1	GetDefinitionURIs request parameters.....	12
7.2.2	GetDefinitionURIs request KVP encoding (mandatory).....	13
7.2.3	GetDefinitionURIs request XML encoding (mandatory).....	13
7.3	GetDefinitionURIs operation response .....	14
7.3.1	Normal response parameters.....	14
7.3.2	Normal response XML encoding.....	15
7.3.3	GetDefinitionURIs response example .....	15
8	GetDefinition operation (mandatory) .....	17
8.1	Introduction .....	17
8.2	GetDefinition operation request .....	17
8.2.1	GetDefinition request parameters .....	17
8.2.2	GetDefinition request KVP encoding (mandatory) .....	17
8.2.3	GetDefinition request XML encoding (mandatory) .....	18
8.2.4	GetDefinition request RESTful encoding (optional) .....	19
8.3	GetDefinition operation response.....	20
8.3.1	Normal response parameters.....	20
8.3.2	Normal response XML encoding.....	20
8.3.3	GetDefinition response example.....	21
9	GetMatchingDefinitions operation (mandatory).....	23
9.1	Introduction .....	23
9.2	GetMatchingDefinitions operation request .....	23
9.2.1	GetMatchingDefinitions request parameters .....	23
9.2.2	GetMatchingDefinitions request KVP encoding (mandatory) .....	24
9.2.3	GetMatchingDefinitions request XML encoding (mandatory).....	24
9.3	GetMatchingDefinitions operation response.....	26

9.3.1	Normal response parameters.....	26
9.3.2	Normal response XML encoding.....	26
9.3.3	GetMatchingDefinitions response example.....	27
10	InsertDefinition operation (mandatory).....	28
10.1	Introduction.....	28
10.2	InsertDefinition operation request.....	28
10.2.1	InsertDefinition request parameters.....	28
10.2.2	InsertDefinition request XML encoding (mandatory).....	28
10.3	InsertDefinition operation response.....	30
10.3.1	Normal response parameters.....	30
10.3.2	Normal response XML encoding.....	31
10.3.3	InsertDefinition response example.....	31
11	DeleteDefinition operation (mandatory).....	33
11.1	Introduction.....	33
11.2	DeleteDefinition operation request.....	33
11.2.1	DeleteDefinition request parameters.....	33
11.2.2	DeleteDefinition request XML encoding (mandatory).....	33
11.3	DeleteDefinition operation response.....	34
11.3.1	Normal response parameters.....	34
11.3.2	Normal response XML encoding.....	35
11.3.3	DeleteDefinition response example.....	35
12	Implementation and Practical Application.....	37
12.1	Implementation.....	37
12.2	Example.....	37
12.2.1	Using the SOR as an Observable Dictionary.....	37
12.2.2	Using the SOR for Discovering Semantically Related Observables.....	38
12.2.3	Integrating the SOR with a Sensor Registry/Catalogue.....	39

<b>Figures</b>	<b>Page</b>
<b>Figure 1: SOR Capabilities shown in XMLSpy notation.....</b>	<b>8</b>
<b>Figure 2: GetDefinitionURIs operation request shown in XMLSpy notation. ....</b>	<b>13</b>
<b>Figure 3: GetDefinitionURIs operation response shown in XMLSpy notation. ....</b>	<b>15</b>
<b>Figure 4: GetDefinition operation request shown in XMLSpy notation. ....</b>	<b>18</b>
<b>Figure 5: GetDefinition operation response shown in XMLSpy notation. ....</b>	<b>21</b>
<b>Figure 6: GetMatchingDefinitions operation request shown in XMLSpy notation. ....</b>	<b>25</b>
<b>Figure 7: GetMatchingDefinitions operation response shown in XMLSpy notation. ....</b>	<b>26</b>
<b>Figure 8: InsertDefinition operation request shown in XMLSpy notation. ....</b>	<b>29</b>
<b>Figure 9: InsertDefinition operation response shown in XMLSpy notation. ....</b>	<b>31</b>
<b>Figure 10: DeleteDefinition operation request shown in XMLSpy notation. ....</b>	<b>33</b>
<b>Figure 11: DeleteDefinition operation response shown in XMLSpy notation. ....</b>	<b>35</b>
<b>Figure 12: Screenshot showing a GetDefinition operation response .....</b>	<b>38</b>
<b>Figure 13: Screenshot showing a GetMatchingDefinitions operation response .....</b>	<b>39</b>
<b>Figure 14: Integration of the SOR into a Sensor Registry/Catalogue .....</b>	<b>40</b>

<b>Tables</b>	<b>Page</b>
<b>Table 1 — Implementation of parameters in GetCapabilities operation request .....</b>	<b>6</b>
<b>Table 2 — Additional Section name values and their meanings .....</b>	<b>7</b>
<b>Table 3 — Parameters included in Contents section.....</b>	<b>7</b>
<b>Table 4 — Parameters in GetDefinitionURIs operation request .....</b>	<b>12</b>
<b>Table 5 — GetDefinitionURIs operation request URL parameters .....</b>	<b>13</b>
<b>Table 6 — Parts of GetDefinitionURIs operation response .....</b>	<b>15</b>
<b>Table 7 — Parameters in GetDefinition operation request .....</b>	<b>17</b>
<b>Table 8 — GetDefinition operation request URL parameters .....</b>	<b>17</b>
<b>Table 9 — Parts of GetDefinition operation response .....</b>	<b>20</b>
<b>Table 10 — Parameters in GetMatchingDefinitionsRequest operation request .....</b>	<b>23</b>
<b>Table 11 — GetMatchingDefinitions operation request URL parameters .....</b>	<b>24</b>
<b>Table 12 — Parts of GetMatchingDefinitions operation response .....</b>	<b>26</b>
<b>Table 13 — Parameters in InsertDefinition operation request.....</b>	<b>28</b>
<b>Table 14 — Parts of InsertDefinition operation response .....</b>	<b>31</b>
<b>Table 15 — Parameters in DeleteDefinition operation request.....</b>	<b>33</b>
<b>Table 16 — Parts of DeleteDefinition operation response .....</b>	<b>35</b>

## **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.

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

## **iii. Submitting organizations**

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

- a. Institute for Geoinformatics (IfGI), University of Münster (WWU)
- b. 52° North Initiative for Geospatial Open Source Software GmbH

## **iv. Document contributor contact points**

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

<b>Name</b>	<b>Organization</b>
Simon Jirka	IfGI, University of Münster
Arne Bröring	52° North GmbH
Daniel Nüst	IfGI, University of Münster

## v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2009-09-05	0.3.0	Simon Jirka	All	First version of the Discussion Paper
2010-09-01	0.4.0	Simon Jirka	All	Updated version including new operations and bindings.
2010-10-08		Carl Reed	Various	Prepare for publication as DP

## vi. Changes to the OGC Abstract Specification

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

## vii. Future work

Improvements in this document are desirable to extend the functionality of the SOR. This includes especially the following aspects:

The SOR addresses the access to ontologies and the execution of reasoning mechanisms. For both of these aspects there might be a more general interest across multiple domains. Thus, it should be investigated how the access to ontologies and the execution of reasoning processes could be encapsulated by separate service interfaces. The pragmatic approach of the SOR can serve as a basis for the approach presented by Janowicz et al. 2009 who propose semantic enablement architecture for SDIs by introducing two new interfaces, the Web Ontology Service (WOS) and highly flexible Web Reasoning Service (WRS). The WOS should be designed as a profile of the CS-W specification (OGC 07-006r1) while the WRS encapsulates semantic web reasoners and should be developed as a profile of the WPS (OGC 05-007r7).

In its current state, the SOR serves only simple reasoning functionality. It allows investigating the hierarchical relationships between phenomena by basic subsumption reasoning. This reasoning capability enables to determine generalization, specialization and equivalency between phenomena. However, a higher degree of freedom and flexibility regarding reasoning abilities is desirable. In a future approach, reasoning should include non-standard inference such as finding the most specific concept, the least common subsumer as well as similarity reasoning.

Additionally, the SOR's focus on handling phenomena and their definitions can be broadened in future. It shall be assessed how the SOR concept could be transferred to further application domains.

## Foreword

This Discussion paper introduces the Sensor Observable Registry (SOR), a web service interface for managing the definitions of phenomena measured by sensors as well as exploring semantic relationships between these phenomena.

With this Discussion Paper the authors aim at presenting a pragmatic solution that may serve as the basis for further discussion and could serve as input to the creation of a more generic solution for handling semantics within the context of sensor networks or even beyond this domain.

This work was supported by the EC projects OSIRIS (<http://www.osiris-fp6.eu/>, contract number 033475) and GENESIS (<http://genesis-fp7.eu/>, contract number 223996) co-funded by the Information Society and Media DG of the European Commission.

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



## Introduction

This Discussion Paper introduces the concept of the Sensor Observable Registry (SOR) which was created within the EC project OSIRIS and further evolved in the EC project GENESIS.

In sensor networks there is a need for handling semantics. This is especially true for handling phenomenon definitions. It is necessary to be able to specify the phenomena that are observed by a sensor so that the discovery of sensors (searching for sensors that observe certain phenomena) and the interpretation of sensor data (requires determining which phenomenon was observed) become possible.

Within the OSIRIS project an important need was to create a solution for managing the phenomena observed by sensors as well as the definitions and semantics of these phenomena. Basically two requirements lead to the development of the SOR:

- The need for accessing definitions of phenomena identified by certain URIs
- The enhancement of the sensor discovery process by exploring and investigating the semantics of observed phenomena

Based on the two above mentioned requirements, the following use cases are supported by the SOR:

- A user wants to search for sensors measuring a certain phenomenon. How can he specify the phenomenon he is interested in? Usually the phenomena measured by sensors are identified by certain URIs. However, as these URIs may vary and as they might even be unknown to the user, he needs some kind of dictionary to look up the available phenomena and their corresponding URIs.
- If users receive metadata about sensor observations, the measured phenomena are identified through URIs. To understand the meaning of these URIs a dictionary is needed that is capable of resolving the URIs and returning according definitions and explanations.
- Across different domains or even within a single domain the same phenomenon might be identified by different URIs. Thus, for being able to find all phenomena which might be of interest to a user, there must be a solution that is capable of determining all URIs identifying the same phenomenon. The foundation for achieving this aim is the exploitation of the semantics that lie behind the URIs which identify a phenomenon and its relation to other phenomena.
- When searching for a sensor, users sometimes are interested in finding sensors that observe a phenomenon which belongs to a broader thematic field or category

of phenomena. For example in case of an accident in a chemical plant a cloud of air pollutants is released. Directly after the accident users do not know its composition. Thus, they need various kinds of air pollutant sensors to determine which chemical substances are contained within the pollutant cloud. One approach might be that the user relies on his knowledge to search for sensors measuring those kinds of air pollutants he knows (e.g. searching separately for sensors measuring CO, NO<sub>2</sub>, SO<sub>2</sub>, H<sub>2</sub>CO, etc.). This approach for searching sensors would be very cumbersome and embodies the risk of forgetting to search for one or more relevant sensor types. Instead a much more efficient approach is to search for sensors measuring the generic phenomenon “air pollution”. In this case one search request would return directly all sensor types the user needs. To make such a type of discovery possible, it is necessary to exploit the semantics of the phenomena measured by sensors. A sensor discovery solution has to be able to determine that CO, NO<sub>2</sub>, SO<sub>2</sub>, H<sub>2</sub>CO, etc. all are some kind of air pollutant. This functionality is provided through the SOR.

# **OGC Sensor Observable Registry Discussion Paper**

## **1 Scope**

This OGC Discussion Paper describes the Sensor Observable Registry (SOR), a web service interface for providing definitions of phenomena that are observed by sensors, for resolving URIs identifying these definitions and for exploring semantic relationships between phenomena.

This document is intended to stimulate the discussion on ways for handling the semantics of sensor observables. Furthermore the basic concepts of the SOR may be advanced in the future so that not only sensor observables but any kind of definition can be handled.

## 2 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.

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

OGC 07-000, *OpenGIS® Sensor Model Language (SensorML) Implementation Specification*

OGC 07-036, *OpenGIS® Geography Markup Language (GML) Encoding Standard*

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

### 3 Terms and definitions

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

**observable**

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

**phenomenon**

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

**sensor**

An entity capable of observing a phenomenon and returning an observed value. [OGC07-000]

**sensor discovery**

Process of searching sensors or SWE services that encapsulate sensors.

## 4 Conventions

### 4.1 Abbreviated terms

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

Some more frequently used abbreviated terms:

SensorML	Sensor Model Language
SIR	Sensor Instance Registry
SOR	Sensor Observable Registry
SWE	Sensor Web Enablement
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name

## 5 Sensor Observable Registry Overview

The specified Sensor Observable Registry is capable of providing definitions of phenomena that are observed by sensors, of resolving URIs identifying these definitions, and of exploring semantic relationships between phenomena.

The SOR interface (currently) specifies six operations that can be requested by a client and performed by a SOR 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 standard version being used for client-server interactions.
- b) **GetDefinitionURIs** (required implementation by servers) – This operation allows a client to retrieve a list of URIs identifying definitions of phenomena that are contained in a specific SOR instance. Optionally, a client is able to specify a text string to search for URIs containing it. Furthermore a paging mechanism is supported for retrieving subsets of a longer list of URIs. This functionality is especially intended for supporting the development of graphical user interfaces as it allows creating lists of URIs for interactive selection.
- c) **GetDefinition** (required implementation by servers) – This operation allows a client to retrieve the definition of an observable identified by a given URI. This definition is returned as a SWE Common Phenomenon.
- d) **GetMatchingDefinitions** (required implementation by servers) – This operation allows a client to retrieve the URIs identifying definitions of observables which are related to another given phenomenon in a certain way. Currently the SOR supports the relations “generalization”, “specialization” and “equivalency”.
- e) **InsertDefinition** (required implementation by servers) – This operation allows a client to insert a new definition of an observable into a SOR instance.
- f) **DeleteDefinition** (required implementation by servers) – This operation allows a client to delete a definition of an observable from a SOR instance.

Many of these interface aspects are common with other OWSs and thus specified in the OpenGIS<sup>®</sup> Web Services Common Implementation Specification [OGC 06-121r3]. Hence, many of the common aspects are normatively referenced herein, instead of being repeated in this standard.

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

## 6 GetCapabilities operation (mandatory)

### 6.1 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. This clause specifies the XML document that a server shall return to describe its capabilities.

### 6.2 Operation request

The GetCapabilities operation request shall be as specified in Subclauses 7.2 and 7.3 of [OGC 06-121r3]. The value of the “service” parameter shall be “SOR”. 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 06-121r3].

The “Multiplicity and use” column in Table 1 of [OGC 06-121r3] specifies the optionality of each listed parameter in the GetCapabilities operation request. Table 1 specifies the implementation of those parameters by SOR clients and servers.

**Table 1 — Implementation of parameters in GetCapabilities operation request**

Names	Multiplicity	Client implementation	Server implementation
service	One (mandatory)	Each parameter shall be implemented by all clients, using specified value	Each parameter shall be implemented by all servers, checking that each parameter is received with specified value
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) <sup>b</sup>	Each parameter may be implemented by each client <sup>b</sup>	Each parameter may be implemented by each server <sup>a</sup>
updateSequence	Zero or one (optional) <sup>b</sup>	If parameter not provided, shall expect default response	If parameter not implemented or not received, shall provide default response
AcceptFormats	Zero or one (optional) <sup>b</sup>	If parameter provided, shall allow default or specified response	If parameter implemented and received, shall provide specified response
<p>a A specific OWS is allowed to make mandatory server implementation of any of these three parameters.</p> <p>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.</p>			

All SOR 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 SOR capabilities document, a client could issue the following KVP encoded GetCapabilities operation request with near-minimum contents:



`http://giv-genesis.uni-muenster.de:8080/SOR/sor?service=SOR&request=getCapabilities`

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

```
<?xml version="1.0" encoding="UTF-8"?>
<sor:GetCapabilities service="SOR"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
  xsi:schemaLocation="http://swsl.uni-
    muenster.de/sor http://giv-genesis.uni-
    muenster.de/schemas/sor/sorAll.xsd"/>
```

### 6.3 GetCapabilities operation response

The service metadata document shall contain the sections that are defined in the OWS Common specification. For the SOR these sections are not modified. The Contents subsection is specific for the SOR. 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.

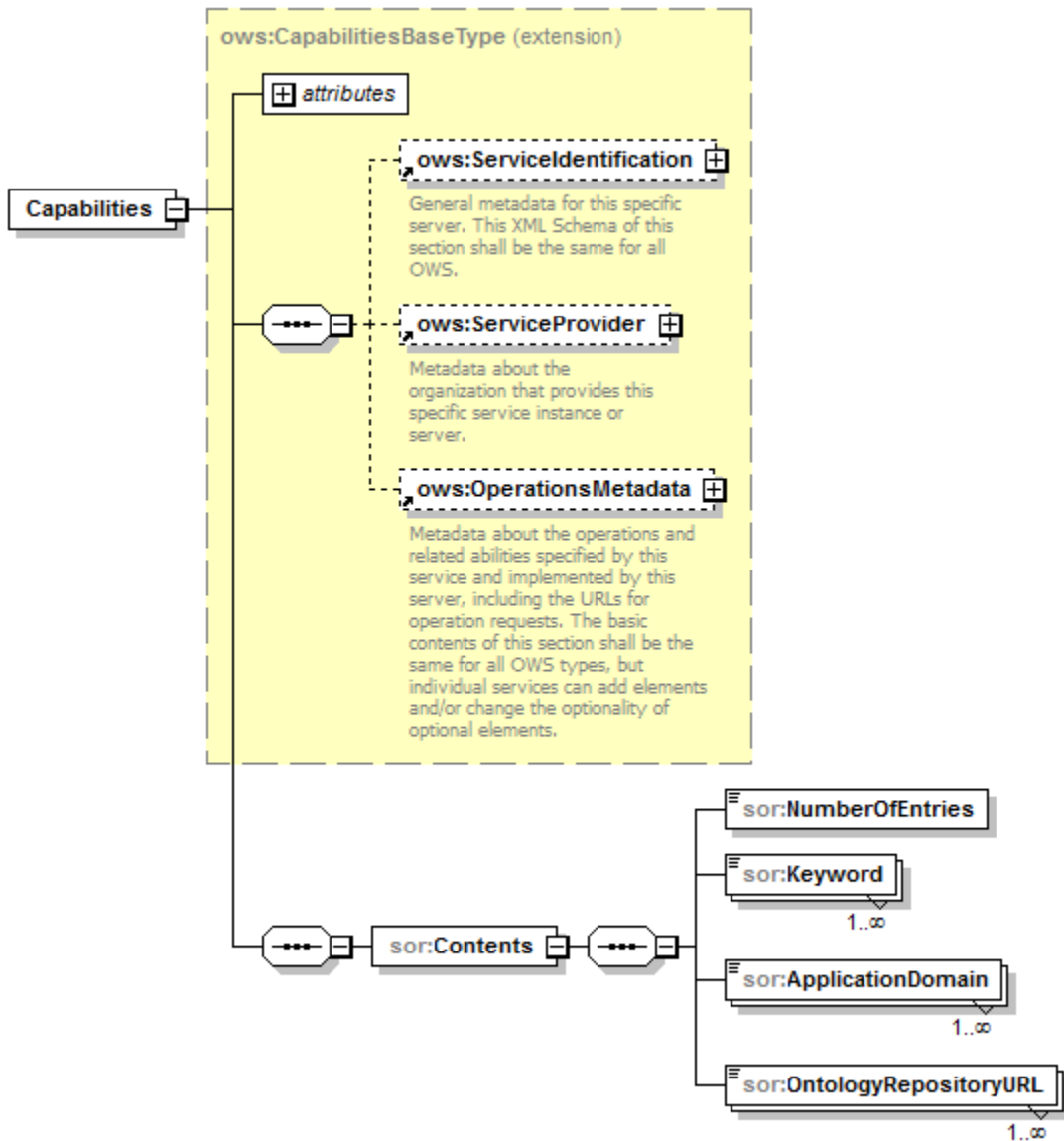
**Table 2 — Additional Section name values and their meanings**

Section name	Contents
Contents	Metadata about the data served by this server. For the SOR, this section shall contain data about the number of entries it contains, human readable keywords as well as application domains it covers and references to the used ontology repositories.

The Contents section of a service metadata document contains metadata about the data served by this server. For the SOR, this Contents section shall contain data about the number of entries it contains, human readable keywords as well as application domains it covers and references to the used ontology repositories. The Contents section shall include the parameters specified in Table 3.

**Table 3 — Parameters included in Contents section**

Names	Definition	Data type and values	Multiplicity and use
NumberOfEntries	Number of entries contained in the SOR instance	Integer	One (mandatory)
Keyword	Keyword describing the content of the SOR instance.	Character string representing a keyword describing the content of the SOR instance	One or more (mandatory)
ApplicationDomain	Description of the application domain to which the SOR can be applied.	Character string representing the name of an application domain	One or more (mandatory)
OntologyRepository URL	URL pointing to a repository containing an ontology used by the SOR instance.	Character string representing a URL	One or more (mandatory)



**Figure 1: SOR Capabilities shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of a SOR Capabilities document which is always encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:sor="http://sws1.uni-muenster.de/sor"
  targetNamespace="http://sws1.uni-muenster.de/sor"
  elementFormDefault="qualified" version="1.0.0" xml:lang="en">
  <import namespace="http://www.opengis.net/ows/1.1"
    schemaLocation=
      "http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
```

```

<element name="Capabilities">
  <complexType>
    <complexContent>
      <extension base="ows:CapabilitiesBaseType">
        <sequence>
          <element name="Contents">
            <complexType>
              <sequence>
                <element name="NumberOfEntries" type="int"/>
                <element name="Keyword" type="string"
                  maxOccurs="unbounded"/>
                <element name="ApplicationDomain"
                  type="string"
                  maxOccurs="unbounded"/>
                <element name="OntologyRepositoryURL"
                  type="anyURI"
                  maxOccurs="unbounded"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
</schema>

```

An example for a SOR Capabilities document:

```

<?xml version="1.0" encoding="UTF-8"?>
<sor:Capabilities xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  version="0.3.0"
  updateSequence="2010-01-01T00:00:00+0100"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  <ows:ServiceIdentification
    xmlns:ows="http://www.opengis.net/ows/1.1"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <ows:Title>WWU SOR</ows:Title>
    <ows:Abstract>SOR Server Operated by WWU</ows:Abstract>
    <ows:ServiceType codeSpace="http://giv-genesis.uni-
      muenster.de:8080/SOR">OGC:SOR</ows:ServiceType>
    <ows:ServiceTypeVersion>0.3.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider xmlns:ows="http://www.opengis.net/ows/1.1"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <ows:ProviderName>
      SWSL - Sensor Web and Simlulation Lab
    </ows:ProviderName>
    <ows:ProviderSite xlink:href="http://giv-genesis.uni-
      muenster.de:8080/SOR"/>
    <ows:ServiceContact>
      <ows:IndividualName>Simon Jirka</ows:IndividualName>

```

```

    <ows:ContactInfo>
      <ows:Phone>
        <ows:Voice>222-222-2222</ows:Voice>
        <ows:Facsimile>333-333-3333</ows:Facsimile>
      </ows:Phone>
    </ows:ContactInfo>
  </ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
        <ows:Post xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
      </ows:HTTP>
    </ows:DCP>
    <ows:Parameter name="Sections">
      <ows:AllowedValues>
        <ows:Value>ServiceIdentification</ows:Value>
        <ows:Value>ServiceProvider</ows:Value>
        <ows:Value>OperationsMetadata</ows:Value>
        <ows:Value>Contents</ows:Value>
        <ows:Value>All</ows:Value>
      </ows:AllowedValues>
    </ows:Parameter>
  </ows:Operation>
  <ows:Operation name="GetDefinitionURIs">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
        <ows:Post xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetDefinition">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
        <ows:Post xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetMatchingDefinitions">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
        <ows:Post xlink:href="http://giv-genesis.uni-
          muenster.de:8080/SOR/sor"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationsMetadata>

```

```

</ows:Operation>
<ows:Operation name="InsertDefinition">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://giv-genesis.uni-
        muenster.de:8080/SOR/sor"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="DeleteDefinition">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://giv-genesis.uni-
        muenster.de:8080/SOR/sor"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Parameter name="service">
  <ows:AllowedValues>
    <ows:Value>SOR</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
<ows:Parameter name="version">
  <ows:AllowedValues>
    <ows:Value>0.3.0</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
</ows:OperationsMetadata>
<sor:Contents>
  <sor:NumberOfEntries>92</sor:NumberOfEntries>
  <sor:Keyword>weather</sor:Keyword>
  <sor:Keyword>climate</sor:Keyword>
  <sor:Keyword>meteorology</sor:Keyword>
  <sor:ApplicationDomain>meteorology</sor:ApplicationDomain>
  <sor:OntologyRepositoryURL>
    http://sweet.jpl.nasa.gov/1.1/
  </sor:OntologyRepositoryURL>
</sor:Contents>
</sor:Capabilities>

```

### 6.3.1 Exceptions

When a SOR server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 06-121r3]. The allowed exception codes shall include those listed in Table 5 of Subclause 7.4.1 of [OGC 06-121r3], assuming the updateSequence parameter is implemented by the server.

## 7 GetDefinitionURIs operation (mandatory)

### 7.1 Introduction

The GetDefinitionURIs operation allows SOR clients to retrieve a list of URIs identifying definitions that are contained in a specific SOR instance. Optionally, a client is able to specify a text string that shall occur within the returned URIs. Furthermore a paging mechanism is supported for retrieving subsets of a longer list of URIs. This functionality is especially intended for supporting the development of client user interfaces as it allows creating lists from which users can select the URIs of those phenomenon definitions they are interested in.

### 7.2 GetDefinitionURIs operation request

#### 7.2.1 GetDefinitionURIs request parameters

A request to perform the GetDefinitionURIs operation shall include the parameters specified in Table 4.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 06-121r3].

**Table 4 — Parameters in GetDefinitionURIs operation request**

Names <sup>a</sup>	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is “SOR”	One (mandatory)
request	Operation name	Character String type, not empty Value is “GetDefinitionURIs”	One (mandatory)
version	Standard version for operation	Character String type, not empty Value is “0.3.0”	One (mandatory)
MaxNumberOfResults	Number of results (entries) that shall be returned	Integer type, 1 or higher (default is 10)	One (optional)
StartResultElement	Index number of the first result element that shall be returned (of all elements matching a possibly given SearchString)	Integer type, 0 or higher (default is 0)	One (optional)
SearchString	Substring that shall occur within the definition URIs to be returned.	Character String type, not empty	One (optional)

<sup>a</sup> The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].

### 7.2.2 GetDefinitionURIs request KVP encoding (mandatory)

Servers may implement HTTP GET transfer of the GetDefinitionURIs operation request, using KVP encoding. The KVP encoding of the GetDefinitionURIs operation request shall use the parameters specified in Table 5. The parameters listed in Table 5 shall be as specified in Table 4.

**Table 5 — GetDefinitionURIs operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=SOR	Mandatory	Service type identifier
request= GetDefinitionURIs	Mandatory	Operation name
version=0.3.0	Mandatory	Standard and schema version for this operation
MaxNumberOfResults=20	Optional	Number of results (entries) that shall be returned
StartResultElement=0	Optional	Index number (integer) of the first result element that shall be returned
SearchString=temperature	Optional	Substring that shall occur within the definition URIs to be returned.

<sup>a</sup> All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 06-121r3].

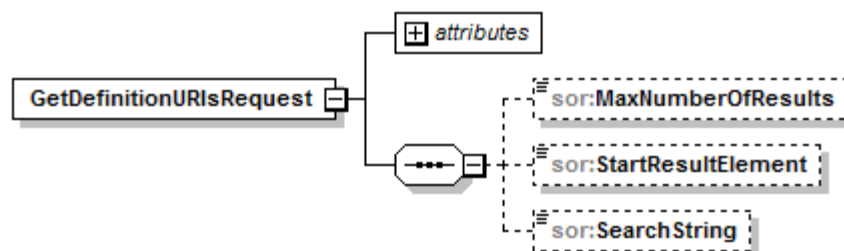
**EXAMPLE** An example GetDefinitionURIs operation request KVP encoded for HTTP GET is:

```
http://giv-genesis.uni-muenster.de:8080/SOR/sor?service=SOR
&request=GetDefinitionURIs&version=0.3.0&maxNumberOfResults=5
&startResultElement=3&searchString=Temperature
```

### 7.2.3 GetDefinitionURIs request XML encoding (mandatory)

All SOR servers shall implement HTTP POST transfer of the GetDefinitionURIs operation request, using XML encoding.

A GetDefinitionURIs operation request is encoded in XML as shown in Figure 2.



**Figure 2: GetDefinitionURIs operation request shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of a GetDefinitionURIs operation request encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <element name="GetDefinitionURIsRequest">
    <complexType>
      <sequence>
        <element name="MaxNumberOfResults" type="int"
          minOccurs="0"/>
        <element name="StartResultElement" type="int"
          minOccurs="0"/>
        <element name="SearchString" type="string" minOccurs="0"/>
      </sequence>
      <attribute name="service" type="string" use="required"
        fixed="SOR"/>
      <attribute name="version" type="string" use="required"
        fixed="0.3.0"/>
    </complexType>
  </element>
</schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<sor:GetDefinitionURIsRequest service="SOR" version="0.3.0"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  <sor:MaxNumberOfResults>3</sor:MaxNumberOfResults>
  <sor:StartResultElement>5</sor:StartResultElement>
  <sor:SearchString>Temperature</sor:SearchString>
</sor:GetDefinitionURIsRequest>

```

## 7.3 GetDefinitionURIs operation response

### 7.3.1 Normal response parameters

The normal response to a valid GetDefinitionURIs operation request shall be a list of URIs identifying observable definitions. More precisely, a response from the GetDefinitionURIs operation shall include the parts listed in Table 6. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

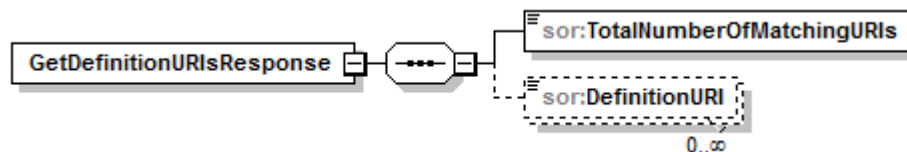


**Table 6 — Parts of GetDefinitionURIs operation response**

Names	Definition	Data type and values	Multiplicity and use
TotalNumberOfMatchingURIs	Number of URIs contained in the SOR that match to the request	Integer type, 0 or higher (default is 0)	One (mandatory)
DefinitionURI	URIs identifying the definitions of phenomena contained in the SOR instance	Character string representing a URI	Zero or more

### 7.3.2 Normal response XML encoding

A GetDefinitionURIs operation response is encoded in XML as shown in Figure 3.



**Figure 3: GetDefinitionURIs operation response shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of a GetDefinitionURIs operation response which is always encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <element name="GetDefinitionURIsResponse">
    <complexType>
      <sequence>
        <element name="TotalNumberOfMatchingURIs" type="int"/>
        <element name="DefinitionURI" type="anyURI" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

### 7.3.3 GetDefinitionURIs response example

A GetDefinitionURIs operation response encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<sor:GetDefinitionURIsResponse xmlns:sor="http://swsl.uni-
```

```

                                muenster.de/sor"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://swsl.uni-muenster.de/sor
                        http://giv-genesis.uni-muenster.de/schemas/sor/
                        sorAll.xsd">
< sor:TotalNumberOfMatchingURIs>7</ sor:TotalNumberOfMatchingURIs>
< sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::Temperature
</ sor:DefinitionURI>
< sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::AirTemperature
</ sor:DefinitionURI>
< sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::WaterTemperature
</ sor:DefinitionURI>
</ sor:GetDefinitionURIsResponse>
```

## 8 GetDefinition operation (mandatory)

### 8.1 Introduction

The GetDefinition operation allows SOR clients to retrieve the definition of an observable identified by a given URI using different methods. This definition is returned (independently from the used method) as a SWE Common Phenomenon.

### 8.2 GetDefinition operation request

#### 8.2.1 GetDefinition request parameters

A request to perform the GetDefinition operation shall include the parameters specified in Table 7.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 06-121r3].

**Table 7 — Parameters in GetDefinition operation request**

Names <sup>a</sup>	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is “SOR”	One (mandatory)
request	Operation name	Character String type, not empty Value is “GetDefinition”	One (mandatory)
version	Standard version for operation	Character String type, not empty Value is “0.3.0”	One (mandatory)
InputURI	URI of the phenomenon for which a definition (dictionary) entry shall be retrieved	Character string representing an URI	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].

#### 8.2.2 GetDefinition request KVP encoding (mandatory)

Servers may implement HTTP GET transfer of the GetDefinition operation request, using KVP encoding. The KVP encoding of the GetDefinition operation request shall use the parameters specified in Table 8. The parameters listed in Table 8 shall be as specified in Table 7.

**Table 8 — GetDefinition operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=SOR	Mandatory	Service type identifier
request= GetDefinition	Mandatory	Operation name
version=0.3.0	Mandatory	Standard and schema version for this operation

InputURI= urn:ogc:def:phenomenon: OGC::Temperature	Mandatory	URI of the phenomenon for which a definition (dictionary) entry shall be retrieved
a All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 06-121r3].		

EXAMPLE An example GetDefinition operation request KVP encoded for HTTP GET is:

```
http://giv-genesis.uni-muenster.de:8080/SOR/sor?service=SOR
&request=GetDefinition&version=0.3.0
&inputURI=urn:ogc:def:phenomenon:OGC::Temperature
```

### 8.2.3 GetDefinition request XML encoding (mandatory)

All SOR servers shall implement HTTP POST transfer of the GetDefinition operation request, using XML encoding.

A GetDefinition operation request is encoded in XML as shown in Figure 4.

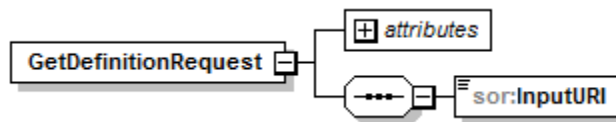


Figure 4: GetDefinition operation request shown in XMLSpy notation.

The following schema fragment specifies the contents and structure of a GetDefinition operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:ns1="http://www.opengis.net/swe/1.0.1"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/swe/1.0.1"
    schemaLocation=
      "http://schemas.opengis.net/sweCommon/1.0.1/phenomenon.xsd"/>
  <element name="GetDefinitionRequest">
    <complexType>
      <sequence>
        <element name="InputURI" type="anyURI"/>
      </sequence>
      <attribute name="service" type="string" use="required"
        fixed="SOR"/>
      <attribute name="version" type="string" use="required"
        fixed="0.3.0"/>
    </complexType>
  </element>
</schema>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<sor:GetDefinitionRequest version="0.3.0" service="SOR"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
  xsi:schemaLocation="http://swsl.uni-
    muenster.de/sor http://giv-genesis.uni-
    muenster.de/schemas/sor/sorAll.xsd">
  <sor:InputURI>urn:ogc:def:phenomenon:OGC::Temperature</sor:InputURI>
</sor:GetDefinitionRequest>
```

#### 8.2.4 GetDefinition request RESTful encoding (optional)

The access to phenomenon definitions is possibly the most common task for many client applications. In order to offer easily resolvable identifiers and references to phenomenon definitions, a SOR server shall also support a RESTful binding (i.e. following the guidelines of Representational State Transfer, in short REST) of the GetDefinition request. Within REST every resource has its own URL via which it can be accessed. This URL shall be structured as follows (parts within < and > are placeholders) and loosely follows the OGC URN Policy (definition branch)<sup>1</sup>:

```
<PROTOCOL_AND_SERVER_ADDRESS>/REST/observable/<AUTHORITY>/
<NAME_OF_OBSERVABLE>/<VERSION (optional)>
```

The URL consists of six elements (separated by forward dashes):

- the root URL of the SIR server
- the fixed keyword “REST” to indicate a RESTful interface type
- the fixed keyword “observable” to indicate the requested object type (allows other resource types on this level in possible future releases)
- the name of the authority maintaining the description of the observable
- the name of the actual observable (unique within one authority and version)
- an optional version parameter

EXAMPLE An example GetDefinition operation request (RESTful encoding, the service URL is “http://giv-genesis.uni-muenster.de/SOR/”):

```
http://giv-genesis.uni-
muenster.de:8080/SOR/REST/observable/OGC/Precipitation/1.0
```

---

<sup>1</sup> <http://www.opengeospatial.org/ogcUrnPolicy>

### 8.3 GetDefinition operation response

#### 8.3.1 Normal response parameters

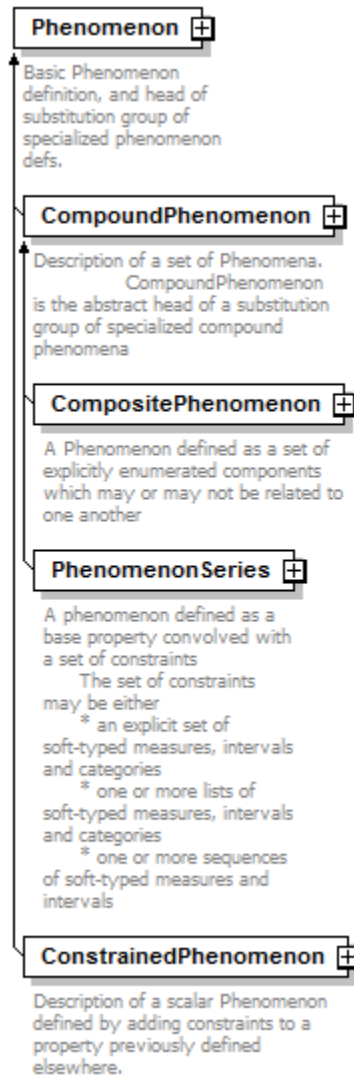
The normal response to a valid GetDefinition operation request shall be a SWE Common Phenomenon containing the description corresponding to the URI that was submitted in the request. More precisely, a response from the GetDefinition operation shall include the parts listed in Table 9. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

**Table 9 — Parts of GetDefinition operation response**

Names	Definition	Data type and values	Multiplicity and use
Phenomenon	Dictionary entry containing the description of a phenomenon, a reference to an ontology, the identifier as well as the name of the phenomenon.	SWE Common Phenomenon	One (mandatory)

#### 8.3.2 Normal response XML encoding

A GetDefinition operation response is encoded in XML as shown in Figure 5.



**Figure 5: GetDefinition operation response shown in XMLSpy notation.**

A GetDefinition operation response is formed by a SWE Common Phenomenon as it is defined in the SWE Common specification.

### 8.3.3 GetDefinition response example

A GetDefinition operation response encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<swe:Phenomenon gml:id="Temperature"
  xsi:schemaLocation="http://www.opengis.net/swe/1.0/gml32 http://giv-genesis.uni-muenster.de/schemas/sweCommon/1.0.1_gml32/phenomenon.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<gml:description xlink:href="http://sweet.jpl.nasa.gov/1.1/
    property.owl#Temperature">
    The degree of hotness or coldness of a body or environment
    (corresponding to its molecular activity)
</gml:description>
<gml:identifier codeSpace="urn:ogc:def:phenomenon:OGC::">
    urn:ogc:def:phenomenon:OGC::Temperature
</gml:identifier>
<gml:name>Temperature</gml:name>
</swe:Phenomenon>
```



## 9 GetMatchingDefinitions operation (mandatory)

### 9.1 Introduction

The GetMatchingDefinitions operation allows SOR clients to retrieve the URIs of definitions of observables which are related to another given phenomenon in a certain way. Currently, the SOR supports the relations “generalization”, “specialization” and “equivalency”.

### 9.2 GetMatchingDefinitions operation request

#### 9.2.1 GetMatchingDefinitions request parameters

A request to perform the GetMatchingDefinitions operation shall include the parameters specified in Table 10.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 06-121r3].

**Table 10 — Parameters in GetMatchingDefinitionsRequest operation request**

Names <sup>a</sup>	Definition	Data type and values	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is “SOR”	One (mandatory)
request request	Operation name	Character String type, not empty Value is “GetMatchingDefinitionsRequest”	One (mandatory)
version version	Standard version for operation	Character String type, not empty Value is “0.3.0”	One (mandatory)
InputURI	URI of the phenomenon for which related phenomena shall be retrieved	Character string representing a URI	One (mandatory)
MatchingType	Type of relation between phenomena that shall be used for reasoning and finding matching phenomena	Character string Value can be “SUPER_TYPE”, “EQUIVALENT_TYPE” or “SUB_TYPE”	One (mandatory)
SearchDepth	Number of intermediate step that are allowed in case of transitively related phenomena	Integer type, 0 or higher (default is 0)	Zero or one (optional)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].

### 9.2.2 GetMatchingDefinitions request KVP encoding (mandatory)

Servers may implement HTTP GET transfer of the GetMatchingDefinitions operation request, using KVP encoding. The KVP encoding of the GetMatchingDefinitions operation request shall use the parameters specified in Table 11. The parameters listed in Table 11 shall be as specified in Table 10 above.

**Table 11 — GetMatchingDefinitions operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=SOR	Mandatory	Service type identifier
request=GetMatchingDefinitionsRequest	Mandatory	Operation name
version=0.3.0	Mandatory	Standard and schema version for this operation
InputURI=urn:ogc:def:phenomenon:OGC::Temperature	Mandatory	Character string representing a URI
MatchingType= SUB_TYPE	Mandatory	Type of relation between phenomena that shall be used for reasoning and finding matching phenomena
SearchDepth=1	Optional (default is 0)	Number of intermediate step that are allowed in case of transitively related phenomena

<sup>a</sup> All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 06-121r3].

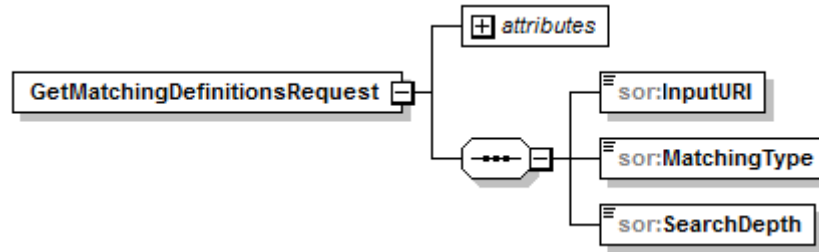
**EXAMPLE** An example GetMatchingDefinitions operation request KVP encoded for HTTP GET is:

```
http://giv-genesis.uni-muenster.de:8080/SOR/sor?service=SOR
&request=GetMatchingDefinitions&version=0.3.0
&inputURI=urn:ogc:def:phenomenon:OGC::Temperature
&matchingType=SUB_TYPE&searchDepth=1
```

### 9.2.3 GetMatchingDefinitions request XML encoding (mandatory)

All SOR servers shall implement HTTP POST transfer of the GetMatchingDefinitions operation request, using XML encoding.

A GetMatchingDefinitions operation request is encoded in XML as shown in Figure 6.



**Figure 6: GetMatchingDefinitions operation request shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of a GetMatchingDefinitions operation request encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0"
  xmlns:mcd="org.n52.reasoner.JenaReasoner."
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <element name="GetMatchingDefinitionsRequest">
    <complexType>
      <sequence>
        <element name="InputURI" type="anyURI"/>
        <element name="MatchingType">
          <simpleType>
            <restriction base="string">
              <enumeration value="SUPER_TYPE"/>
              <enumeration value="EQUIVALENT_TYPE"/>
              <enumeration value="SUB_TYPE"/>
            </restriction>
          </simpleType>
        </element>
        <element name="SearchDepth">
          <simpleType>
            <restriction base="int">
              <minInclusive value="1"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
      <attribute name="service" type="string" use="required"
        fixed="SOR"/>
      <attribute name="version" type="string" use="required"
        fixed="0.3.0"/>
    </complexType>
  </element>
</schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<sor:GetMatchingDefinitionsRequest version="0.3.0" service="SOR"
  xmlns:sor="http://swsl.uni-muenster.de/sor"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://swsl.uni-muenster.de/sor
http://giv-genesis.uni-muenster.de/schemas/sor/
sorAll.xsd">
<sor:InputURI>urn:ogc:def:phenomenon:OGC::Temperature</sor:InputURI>
<sor:MatchingType>SUB_TYPE</sor:MatchingType>
<sor:SearchDepth>1</sor:SearchDepth>
</sor:GetMatchingDefinitionsRequest>

```

### 9.3 GetMatchingDefinitions operation response

#### 9.3.1 Normal response parameters

The normal response to a valid GetMatchingDefinitions operation request shall be a list of URIs of all observable definitions that match the criteria submitted in the request. More precisely, a response from the GetMatchingDefinitions operation shall include the parts listed in Table 12. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

**Table 12 — Parts of GetMatchingDefinitions operation response**

Names	Definition	Data type and values	Multiplicity and use
DefinitionURI	URIs identifying the definitions of phenomena that matched to the parameters specified in the request	Character string representing a URI	Zero or more

#### 9.3.2 Normal response XML encoding

A GetMatchingDefinitions operation response is encoded in XML as shown in Figure 7.



**Figure 7: GetMatchingDefinitions operation response shown in XMLSpy notation.**

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

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sor="http://swsl.uni-muenster.de/sor"
xmlns:swe="http://www.opengis.net/swe/1.0"
xmlns:mcd="org.n52.reasoner.JenaReasoner."
targetNamespace="http://swsl.uni-muenster.de/sor"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<element name="GetMatchingDefinitionsResponse">
<complexType>
<sequence>

```

```

        <element name="MatchingURI" type="anyURI" minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>
</element>
</schema>

```

### 9.3.3 GetMatchingDefinitions response example

A GetMatchingDefinitions operation response encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<sor:GetMatchingDefinitionsResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sor="http://swsl.uni-muenster.de/sor"
    xsi:schemaLocation="http://swsl.uni-muenster.de/sor
        http://giv-genesis.uni-muenster.de/schemas/sor/
        sorAll.xsd">
    <sor:MatchingURI>
        urn:ogc:def:phenomenon:OGC::AirTemperature
    </sor:MatchingURI>
    <sor:MatchingURI>
        urn:ogc:def:phenomenon:OGC::WaterTemperature
    </sor:MatchingURI>
</sor:GetMatchingDefinitionsResponse>

```

## 10 InsertDefinition operation (mandatory)

### 10.1 Introduction

The InsertDefinition operation allows SOR clients to insert a new definition of an observable into a SOR instance.

### 10.2 InsertDefinition operation request

#### 10.2.1 InsertDefinition request parameters

A request to perform the InsertDefinition operation shall include the parameters specified in Table 13. The given swe:Phenomenon includes the mandatory element gml:identifier which is extracted as the identifier by the SOR. It can be used as InputURI and DefinitionURI in the aforementioned operations GetDefinition and GetMatchingDefinitions respectively.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 06-121r3].

**Table 13 — Parameters in InsertDefinition operation request**

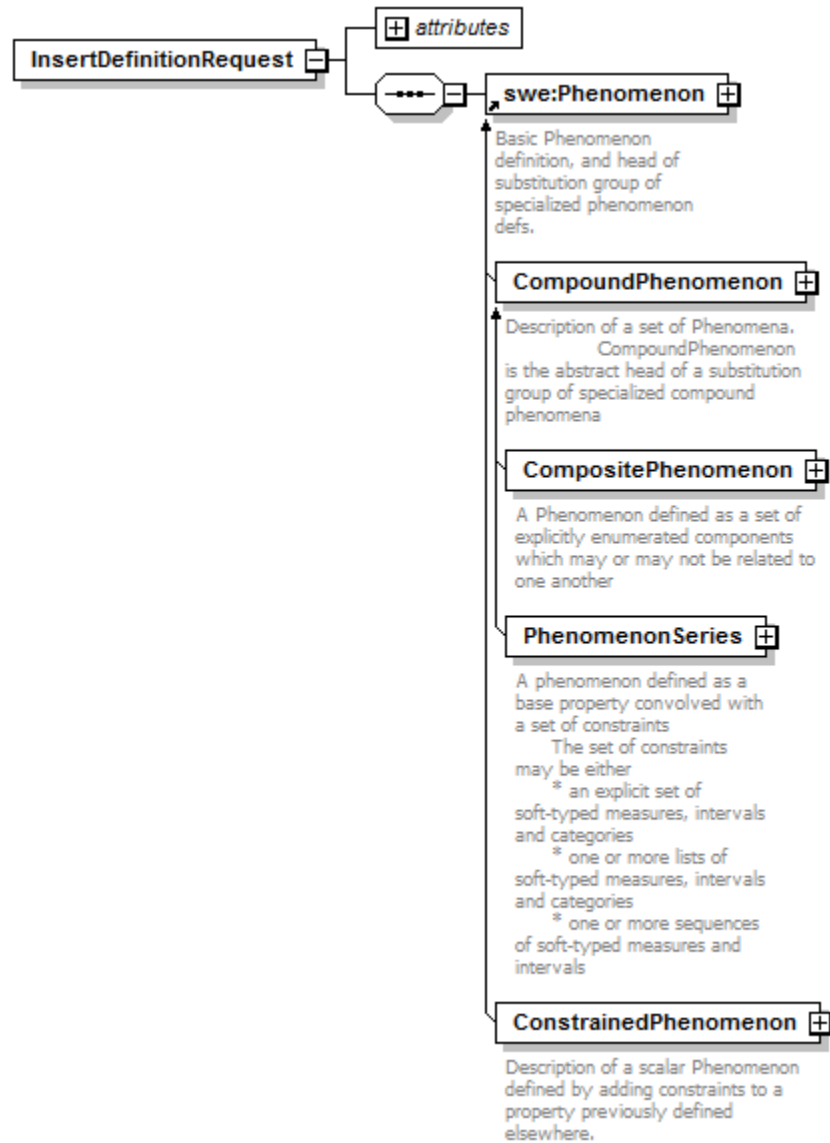
Names <sup>a</sup>	Definition	Data type and values	Multiplicity and use
Service	Service type identifier	Character String type, not empty Value is “SOR”	One (mandatory)
Request	Operation name	Character String type, not empty Value is “InsertDefinition”	One (mandatory)
Version	Standard version for operation	Character String type, not empty Value is “0.3.0”	One (mandatory)
Phenomenon	Dictionary entry containing the description of a phenomenon, a reference to an ontology, the identifier as well as the name of the phenomenon.	SWE Common Phenomenon	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].

#### 10.2.2 InsertDefinition request XML encoding (mandatory)

All SOR servers shall implement HTTP POST transfer of the InsertDefinition operation request, using XML encoding.

An InsertDefinition operation request is encoded in XML as shown in Figure 8.



**Figure 8: InsertDefinition operation request shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of an InsertDefinition operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:ns1="http://www.opengis.net/swe/1.0.1"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/swe/1.0/gml32"
    schemaLocation="http://giv-genesis.uni-muenster.de/schemas
      /sweCommon/1.0.1_gml32/phenomenon.xsd"/>
  <element name="InsertDefinitionRequest">
```

```

    <complexType>
      <sequence>
        <element ref="swe:Phenomenon"/>
      </sequence>
      <attribute name="service" type="string" use="required"
        fixed="SOR"/>
      <attribute name="version" type="string" use="required"
        fixed="0.3.0"/>
    </complexType>
  </element>
</schema>

```

EXAMPLE An example InsertDefinition operation request XML encoded for HTTP POST:

```

<?xml version="1.0" encoding="UTF-8"?>
<sor:InsertDefinitionRequest version="0.3.0" service="SOR"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  <swe:Phenomenon gml:id="Temperature"
    xsi:schemaLocation="http://www.opengis.net/swe/1.0/gml32
    http://giv-genesis.uni-muenster.de/schemas/sweCommon/
    1.0.1_gml32/phenomenon.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <gml:description xlink:href="
      http://sweet.jpl.nasa.gov/1.1/property.owl#Temperature">
      The degree of hotness or coldness of a body or environment
      (corresponding to its molecular activity)
    </gml:description>
    <gml:identifier codeSpace="urn:ogc:def:phenomenon:OGC::">
      urn:ogc:def:phenomenon:OGC::Temperature
    </gml:identifier>
    <gml:name>Temperature</gml:name>
  </swe:Phenomenon>
</sor:InsertDefinitionRequest>

```

## 10.3 InsertDefinition operation response

### 10.3.1 Normal response parameters

The normal response to a valid InsertDefinition operation request shall be the URI of the phenomenon for which the definition (SWE Common Phenomenon) was inserted. More precisely, a response from the InsertDefinition operation shall include the parts listed in Table 14. This table also specifies the UML model data type plus the multiplicity and use of each listed part.



**Table 14 — Parts of InsertDefinition operation response**

Names	Definition	Data type and values	Multiplicity and use
DefinitionURI	URI of the phenomenon for which the definition (dictionary entry) was inserted	Character string representing a URI	One (mandatory)

### 10.3.2 Normal response XML encoding

An InsertDefinition operation response is encoded in XML as shown in Figure 9.



**Figure 9: InsertDefinition operation response shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of an InsertDefinition operation response which is always encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:ns1="http://www.opengis.net/swe/1.0.1"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/swe/1.0/gml32"
    schemaLocation="http://giv-genesis.uni-muenster.de/schemas/
      sweCommon/1.0.1_gml32/phenomenon.xsd"/>
  <element name="InsertDefinitionResponse">
    <complexType>
      <sequence>
        <element name="DefinitionURI" type="anyURI"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

### 10.3.3 InsertDefinition response example

An example for an InsertDefinition operation response encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
< sor:InsertDefinitionResponse
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  < sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::Temperature
  </ sor:DefinitionURI>
```

`</sor:InsertDefinitionResponse>`

## 11 DeleteDefinition operation (mandatory)

### 11.1 Introduction

The DeleteDefinition operation allows SOR clients to delete the definition of an observable identified by a given URI from a SOR instance.

### 11.2 DeleteDefinition operation request

#### 11.2.1 DeleteDefinition request parameters

A request to perform the DeleteDefinition operation shall include the parameters specified in Table 15.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 06-121r3].

**Table 15 — Parameters in DeleteDefinition operation request**

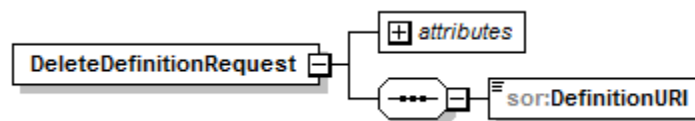
Names <sup>a</sup>	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is “SOR”	One (mandatory)
request	Operation name	Character String type, not empty Value is “DeleteDefinition”	One (mandatory)
version	Standard version for operation	Character String type, not empty Value is “0.3.0”	One (mandatory)
DefinitionURI	URI of the phenomenon for which a definition (dictionary) entry shall be deleted	Character string representing a URI	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r3].

#### 11.2.2 DeleteDefinition request XML encoding (mandatory)

All SOR servers shall implement HTTP POST transfer of the DeleteDefinition operation request, using XML encoding.

A DeleteDefinition operation request is encoded in XML as shown in Figure 10.



**Figure 10: DeleteDefinition operation request shown in XMLSpy notation.**

The following schema fragment specifies the contents and structure of a DeleteDefinition operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:ns1="http://www.opengis.net/swe/1.0.1"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/swe/1.0.1"
    schemaLocation="http://schemas.opengis.net/
      sweCommon/1.0.1/phenomenon.xsd"/>
  <element name="DeleteDefinitionRequest">
    <complexType>
      <sequence>
        <element name="DefinitionURI" type="anyURI"/>
      </sequence>
      <attribute name="service" type="string" use="required"
        fixed="SOR"/>
      <attribute name="version" type="string" use="required"
        fixed="0.3.0"/>
    </complexType>
  </element>
</schema>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<sor:DeleteDefinitionRequest version="0.3.0" service="SOR"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  <sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::Temperature
  </sor:DefinitionURI>
</sor:DeleteDefinitionRequest>
```

## 11.3 DeleteDefinition operation response

### 11.3.1 Normal response parameters

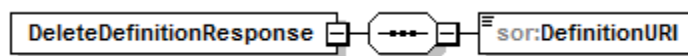
The normal response to a valid DeleteDefinition operation request shall be the URI of the phenomenon for which the definition (dictionary) entry was deleted. More precisely, a response from the DeleteDefinition operation shall include the parts listed in Table 16. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

**Table 16 — Parts of DeleteDefinition operation response**

Names	Definition	Data type and values	Multiplicity and use
DefinitionURI	URI of the phenomenon for which the definition (dictionary) entry was deleted	Character string representing a URI	One (mandatory)

### 11.3.2 Normal response XML encoding

A DeleteDefinition operation response is encoded in XML as shown in Figure 11.



**Figure 11: DeleteDefinition operation response shown in XMLSpy notation.**

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

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:swe="http://www.opengis.net/swe/1.0/gml32"
  xmlns:ns1="http://www.opengis.net/swe/1.0.1"
  targetNamespace="http://swsl.uni-muenster.de/sor"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/swe/1.0.1"
    schemaLocation="http://schemas.opengis.net/sweCommon/1.0.1/
      phenomenon.xsd"/>
  <element name="DeleteDefinitionResponse">
    <complexType>
      <sequence>
        <element name="DefinitionURI" type="anyURI"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

### 11.3.3 DeleteDefinition response example

A DeleteDefinition operation response encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<sor:DeleteDefinitionResponse
  xmlns:sor="http://swsl.uni-muenster.de/sor"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://swsl.uni-muenster.de/sor
    http://giv-genesis.uni-muenster.de/schemas/sor/
    sorAll.xsd">
  <sor:DefinitionURI>
    urn:ogc:def:phenomenon:OGC::Temperature
  </sor:DefinitionURI>
```

```
</sor:DeleteDefinitionResponse>
```

## 12 Implementation and Practical Application

### 12.1 Implementation

Within the OSIRIS project a first prototypical implementation was performed by Thales Communications. In addition, after the end of the project a further going and completely new implementation was created by the University of Münster in the context of the GENESIS project which is made available through the open source initiative 52° North<sup>2</sup>.

For describing the definitions of observables, their URIs and the links to ontologies, the structure of the phenomenon dictionary as defined in the SWE Common specification is used as a basis. This dictionary provides means for mapping URIs to phenomenon definitions and for establishing the link to the according ontologies. The operations GetDefinitionURIs and GetDefinition are realized based on this dictionary.

In order to enable reasoning mechanisms, which form the basis for the GetMarchingDefinitions operation, the Jena Semantic Web Framework<sup>3</sup> is used. The observable definitions were linked to the SWEET ontology<sup>4</sup> to enable execution of reasoning mechanisms.

### 12.2 Example

#### 12.2.1 Using the SOR as an Observable Dictionary

As stated before, one of the core functionalities of the SOR is to resolve URIs identifying the definitions of observables.

Figure 12 shows a screenshot of a web based client accessing an instance of the 52° North SOR. The list of all available URIs in the upper third of the screenshot is generated by the client using the GetDefinitionURIs operation and allows users to select the URIs they want to resolve. After clicking on the button “Build request” the according XML encoding of the GetDefinition operation can be inspected and changed in the text field in the middle of the screen. The request is then send to the server (button “Send request”) and the response is displayed in the text field at the bottom of the screen.

In this example a user has requested the definition of the observable identified by the URI “urn:ogc:def:phenomenon:OGC::Temperature”. The returned definition contains besides the id and name of the definition also a short full text explanation as well as a link to an ontology where the semantics of the observable is formally defined.

---

<sup>2</sup> <https://52north.org/>

<sup>3</sup> <http://jena.sourceforge.net/>

<sup>4</sup> <http://sweet.jpl.nasa.gov/ontology/>

**Get Definition Request:**

Input URI:

```
urn:ogc:def:phenomenon:OGC::RelativeHumidity
urn:ogc:def:phenomenon:OGC::Shape
urn:ogc:def:phenomenon:OGC::Speed
urn:ogc:def:phenomenon:OGC::DiscreteSpectrumTM
urn:ogc:def:phenomenon:OGC::SurfaceWaterTemperature
urn:ogc:def:phenomenon:OGC::Temperature
urn:ogc:def:phenomenon:OGC::BrightnessTemperature
urn:ogc:def:phenomenon:OGC::DeicedTemperature
urn:ogc:def:phenomenon:OGC::DewPointTemperature
urn:ogc:def:phenomenon:OGC::DewPointDepression
```

Build request

```
1 <GetDefinitionRequest service="SOR" version="0.3.0" xmlns="http://swsl.uni-muenster.de
  /sor">
2   <InputURI>urn:ogc:def:phenomenon:OGC::Temperature</InputURI>
3 </GetDefinitionRequest>
4
5
```

Send request

```
1 <swe:Phenomenon gml:id="Temperature" xsi:schemaLocation="http://www.opengis.net/swe/1.0
  /gml32 http://giv-genesis.uni-muenster.de/schemas/sweCommon/1.0.1_gml32/phenomenon.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net
  /gml/3.2" xmlns:swe="http://www.opengis.net/swe/1.0/gml32" xmlns:xlink="http://www.w3.org
  /1999/xlink">
2   <gml:description xlink:href="http://sweet.jpl.nasa.gov/1.1/property.owl#Temperature">The
  degree of hotness or coldness of a body or environment (corresponding to its molecular
  activity)</gml:description>
3   <gml:identifier codeSpace="urn:x-ogc:tc:arch:doc-
  rp(05-010)">urn:ogc:def:phenomenon:OGC::Temperature</gml:identifier>
4   <gml:name>Temperature</gml:name>
5 </swe:Phenomenon>
6
7
8
9
10
11
```

**Figure 12: Screenshot showing a GetDefinition operation response****12.2.2 Using the SOR for Discovering Semantically Related Observables**

The second main function of the SOR is to exploit the semantics of observables in order to find related observables. A basic web based client supporting this functionality is shown in Figure 13.

Like in the previous example the client offers the user a list of the observables' URIs the SOR instance contains. After having selected a URI the user is able to specify the relation that he is interested in (SUB\_TYPE, SUPER\_TYPE, EQUIVALENT\_TYPE) through a drop-down list and a search depth that specifies the number of allowed intermediate steps if transitively related observables shall be returned.



In the example below the user requests sub types (with maximum one intermediate step) of the observable identified by the URI “urn:ogc:def:phenomenon:OGC::Temperature”. The URIs of the matching observable definitions are shown as a list in the lower half of the screenshot.

### Get Matching Definitions Request:

Input URI:

- urn:ogc:def:phenomenon:OGC::RelativeHumidity
- urn:ogc:def:phenomenon:OGC::Shape
- urn:ogc:def:phenomenon:OGC::Speed
- urn:ogc:def:phenomenon:OGC::DiscreteSpectrumTM
- urn:ogc:def:phenomenon:OGC::SurfaceWaterTemperature
- urn:ogc:def:phenomenon:OGC::Temperature
- urn:ogc:def:phenomenon:OGC::BrightnessTemperature
- urn:ogc:def:phenomenon:OGC::DeicedTemperature
- urn:ogc:def:phenomenon:OGC::DewPointTemperature
- urn:ogc:def:phenomenon:OGC::DewPointDepression

Matching Type: SUB\_TYPE

Search Depth: 1

Build request

```

1 <GetMatchingDefinitionsRequest service="SOR" version="0.3.0" xmlns="http://swsl.uni-
2   muenster.de/sor">
3   <InputURI>urn:ogc:def:phenomenon:OGC::Temperature</InputURI>
4   <MatchingType>SUB_TYPE</MatchingType>
5   <SearchDepth>1</SearchDepth>
6 </GetMatchingDefinitionsRequest>

```

Send request

```

4 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::DewPointTemperature</sor:MatchingURI>
5 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::DewPointDepression</sor:MatchingURI>
6 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::DryBulbTemperature</sor:MatchingURI>
7 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::ElectronTemperature</sor:MatchingURI>
8 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::EquivalentTemperature</sor:MatchingURI>
9 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::IonTemperature</sor:MatchingURI>
10 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::NeutralTemperature</sor:MatchingURI>
11 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::PotentialTemperature</sor:MatchingURI>
12 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::PseudoEquivalentPotentialTemperature</sor:Ma
13 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::PseudoEquivalentTemperature</sor:MatchingURI
14 <sor:MatchingURI>urn:ogc:def:phenomenon:OGC::StaticTemperature</sor:MatchingURI>
15

```

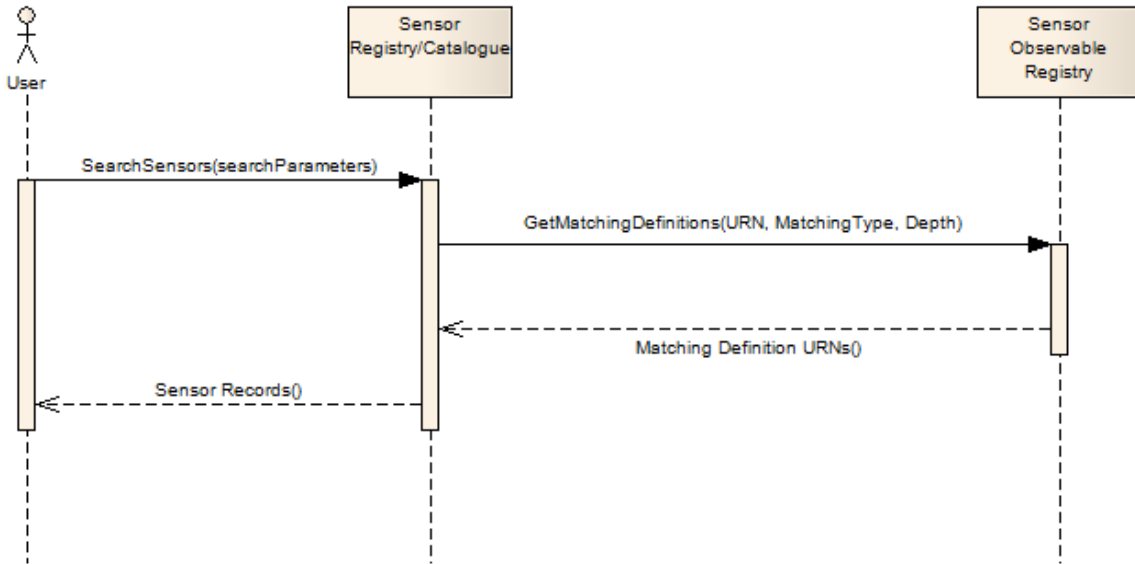
Figure 13: Screenshot showing a GetMatchingDefinitions operation response

### 12.2.3 Integrating the SOR with a Sensor Registry/Catalogue

Finally, this third example shall illustrate how the SOR functionality, especially the GetMatchingDefinitions operation, can be used for improving sensor discovery mechanism. During the OSIRIS project, the enhancement of sensor discovery mechanisms relying on semantics was the driving factor for developing the SOR.

Figure 14 gives an overview how a sensor discovery service (a sensor registry/catalogue) can be combined with the SOR in order to achieve the previously mentioned sensor discovery enhancements.

In the example a user, who searches for a specific kind of sensor, submits his search request to a sensor registry/catalogue. Within this search request the observable for which the user wants the find sensor data is included as a URI. If the user enables the use of the SOR by setting according parameters the sensor registry/catalogue directly queries the SOR for the URIs of those observables which are related to the given observable. After the sensor registry/catalogue has received this list of URIs it queries its internal sensor index for all sensors that measure an observable with any of these URIs. Subsequently registry/catalogue entries for all matching sensors are returned.



**Figure 14: Integration of the SOR into a Sensor Registry/Catalogue**

## Bibliography

- [1] OpenGIS® Sensor Model Language (SensorML) Implementation Specification, OGC document 07-000
- [2] OpenGIS® Geography Markup Language (GML) Encoding Standard, OGC document 07-036
- [3] Janowicz, K.; Kessler, C.; Bröring, A.; Stasch, C.; Schade, S.. 2009. “Towards Semantic Enablement for Spatial Data Infrastructures”. EuroSSC 2009: 4th European Conference on Smart Sensing and Context.
- [4] Jirka, S.; Bröring, A.; Foerster, T.. 2010. “Handling the Semantics of Sensor Observables within SWE Discovery Solutions”. Workshop on Sensor Web Enablement 2010 (SWE2010). In: Proceedings of The 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010).
- [5] Jirka, S.; Bröring, A.; Stasch, C.. 2009. “Discovery Mechanisms for the Sensor Web”. Sensors 9, no. 4: 2661-2681. Online: <http://www.mdpi.com/1424-8220/9/4/2661>
- [6] Lutz, M.; Klien, E.. 2004. “Ontology-based retrieval of geographic information”. Int. J. Geogr. Inf. Sci. 2004, 20, 233-260.
- [7] OSIRIS Consortium. 2008. “Deliverable 6001 Revision A - OSIRIS Architecture Specification and Justification”.
- [8] GENESIS Consortium. 2010. “D5200.1 Report On Sensor Network Architecture Assessment Activity”
- [9] GENESIS Consortium. 2010. “D5200.2 SWE Demonstrator Explicative Notice”
- [10] NASA SWEET ontology (<http://sweet.jpl.nasa.gov/ontology/>)
- [11] Roy T. Fielding. “Architectural styles and the design of network-based software architectures”. PhD Thesis, University of California, Irvine, 2000. Online: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>