Specification of the SWE ingestion service, including SWE profiles and architecture

WP9 - Deliverable D9.9

HORIZON **2020**

sdn-userdesk@seadatanet.org – www.seadatanet.org

SeaDataCloud - Further developing the pan-European infrastructure for marine and ocean data management

Grant Agreement Number: 730960

Deliverable number	Short title
D9.9	Specification of the SWE ingestion service

Long title
Specification of the SWE ingestion service, including SWE profiles and architecture

Short description
This document describes the SeaDataCloud SWE Ingestion Service and several related aspects (i.e. SWE Ingestion Architecture and necessary SWE profiles). The aim of this component is to support sensor operators, researchers and data owners during the publication of collected marine observation data.

Author	Working group
Simon Jirka and Christian Autermann – 52North,	WP 9

Dissemination	Copyright terms
Public	

History

Version	Authors	Date	Comments
0.9	Simon Jirka, Christian Autermann	28/08/2017	Near-final version shared with partners involved in the SWE Ingestion Service
1.0	Simon Jirka, Christian Autermann	31/08/2017	Version submitted to the technical coordinator for review
1.1	Simon Jirka, Christian Autermann	04/10/2017	Version submitted to the technical coordinator for acceptance
1.2	Dick Schaap	12/10/2017	Accepted and submitted to project coordinator for EC submission

1.3	Michèle Fichaut	12/10/2017	Final edits before submission

Table of contents

1. Abbreviations

CDI	Common Data Index
CSV	Comma-separated Values
FTP	File Transfer Protocol
ISO	International Organization for Standardization
O&M	Observations and Measurements
OGC	Open Geospatial Consortium
SEISI	Smart Electronic Interface for Sensor Interoperability
SensorML	Sensor Model Language
SOS	Sensor Observation Service
SSN	Semantic Sensor Network (SSN)
SWE	Sensor Web Enablement
W3C	World Wide Web Consortium

2. Introduction

This document describes the SeaDataCloud SWE Ingestion Service and several related aspects (i.e. SWE Ingestion Architecture and necessary SWE profiles). The aim of this component is to support sensor operators, researchers and data owners during the publication of collected marine observation data.

To enable this ingestion workflow, it is first necessary to describe the data that shall be published as well as the sources that have generated these measurements (e.g. observatories, sensors, gliders, buoys, etc.). If data from such sources shall be consumed automatically (e.g. through data publication modules/importers) it is furthermore necessary to formally specify the specific commands and their parameters offered by a device to retrieve the observed data.

Consequently, this deliverable does not only focus on the SWE Ingestion Service itself. Instead it covers a range of different specifications and components that are necessary for enabling a set of different data publication workflows:

- Devices that are pushing autonomously data to the SWE Ingestion Service
- Devices that shall autonomously be queried by the SWE Ingestion Service to harvest their collected data
- Automatic collection of data from regularly updated repositories (e.g. continuously checking an FTP server for new data files)
- Web-based upload of data files (e.g. CSV) by researchers

To cover this range of functionality, this deliverable introduces a series of different components and specifications that can be used for building a flexible SWE data ingestion workflow. Important elements covered in this deliverable are:

- Specification of a data and metadata upload interface (section 4)
- Specification of an approach to describe the interface of measurement devices (i.e. a dedicated SensorML profile, section 6.1.1)
- Specification of an approach for describing the structure of observation data sets/streams (sections 5.4.2 and 6.1.2)
- Specification of feedback that shall be provided to data publishers (section 7.3)
- Specification of user interface requirements (sections 7.1 and 7.2)

Furthermore, to ensure not only syntactic but also semantic interoperability between data sources, the presented import mechanism will consider the vocabulary developments going on in WP8 and the marine community (especially vocabulary servers operated by BODC).

The development of a first version of the SWE Ingestion Service starts with the completion of this document. Whenever possible, these developments will rely on enhancing existing (open source) software components such as the 52°North SOS importer[1] as well as the SensorML editor smle[2] published by 52°North.

[1] https://github.com/52North/sos-importer

[2] https://github.com/52North/smle

3. Relevant Standards and Inputs

Before describing the specification of the SeaDataCloud SWE Ingestion Service, this section introduces important underlying standards and activities that are relevant inputs.

3.1. OGC Sensor Web Enablement (SWE)

There is a very large amount of different sensor data encodings, data models as well as interfaces. This heterogeneity makes the integration of sensor data a very cumbersome task. For example, without a common standardised approach, it would be necessary to customise each application that shall consume sensor data to the individual data formats and interfaces of all sensing devices that will deliver data.

To address this issue, the Open Geospatial Consortium (OGC), an international de-facto standardisation organisation in the field of spatial information infrastructures has developed the Sensor Web Enablement (SWE) framework of standards. The OGC SWE architecture comprises several specifications facilitating the sharing of observation data and metadata via the Web. Important building blocks are standards for observation data models, for the corresponding metadata about measurement processes, and interfaces for providing sensor-related functionality (e.g. data access) via the World Wide Web.

The following subsections introduce those OGC SWE standards which are relevant for the specification of the SeaDataCloud SWE Ingestion Service.

3.1.1. OGC Sensor Observation Service (SOS)

The OGC Sensor Observation Service (SOS) in its current version 2.0 (Bröring et al., 2012) specifies a Web service interface for querying observation data and metadata.

Besides operations enabling the data download, it also offers upload functionality for publishing observation data. This upload functionality is especially relevant for the specification of the SeaDataCloud SWE Ingestion Service. More details and workflow examples of the data upload functionality is provided in section 5.2.

Within the frame of the European INSPIRE directive, the SOS standard is recommended to be used as Download Service interface for observation data (INSPIRE MIG sub-group MIWP-7a, 2016b).

3.1.2. ISO/OGC Observations and Measurements (O&M)

While the SOS interface addresses functional aspects, there is also a need to specify the data model and format for encoding observation data sets that are for example consumed from or uploaded to an SOS server. This aspect if covered by the ISO/OGC Observations and Measurement 2.0 specification.

This specification consists of two separate parts. The abstract model of the O&M specification is provided as an ISO standards (ISO TC 211, 2011). This ISO standard basically defines the entities, properties and relations that need to consider when modelling observation data. This comprises for example the definition of different types of time stamps or the elements of an individual observation (e.g. observed properties, features of interest, etc.).

This abstract specification is complemented by a XML encoding standard that has been published by the OGC (Cox, 2011). Based on the abstract ISO standard, this OGC specification defines how the different elements of an observation shall be encoded as XML documents.

Within the European INSPIRE directive (see section 3.2), O&M is used as a standardised model and encoding for the provision of different types of observation data (INSPIRE MIG sub-group MIWP-7a, 2016a). This also includes guidance on the application of specific observation types on different kinds of data.

3.1.3. OGC Sensor Model Language (SensorML)

In addition to the observation data itself, users usually need metadata about the process how an observation was generated. Typically this consists of information about the sensor that has generated an observation data set. The specification of a metadata model and XML encoding is the purpose of the OGC Sensor Model Language (SensorML) 2.0 standard (Botts and Robin, 2014).

An overview of important SensorML elements needed for the realisation of the SeaDataCloud SWE Ingestion Service is shown in Table 1.

Based on SensorML metadata, it is on the one hand possible to ensure the correct interpretation of observation data and on the other hand to discover sensors/sensor data sources such as ocean observatories (Delory and Jirka, 2016).

3.2. INSPIRE

INSPIRE (Infrastructure for Spatial Information in the European Community) is a European legal, organizational and technical framework that is aimed at the sharing of geospatial information within the European Community.

The whole INSPIRE framework is based on a directive of the European Union (European Parliament and European Council, 2007) which is complemented by a set of implementing rules and technical guidance documents.

An important principle of the INSPIRE idea is the requirement for European member states to provide access to a defined set of geospatial information through interoperable technologies via the World Wide Web.

As part of the INSPIRE Technical Guidance, the Sensor Web standards SOS and O&M are recommended to share observation data in an interoperable manner.

3.3. Marine SWE Profiles

The OGC SWE standards were designed to be used in a very broad range of different thematic domains. To allow this flexibility, it was necessary to design the SWE standards with a certain degree of freedom in their application (e.g. soft-typed contents, a high amount of optional elements). However, this flexibility comes at the cost of potential interoperability issues. Thus, there is a need for profiles that restrict the domain-independent SWE standards to the specific needs of application domains.

To develop best practice guidance for a profile how to apply the SWE standards in marine projects, several projects on a European and global level are actively cooperating on this topic in a dedicated, informal working group: The Marine SWE Profiles Working Group.

Important topics of this working group comprise:

- Harmonise the application of OGC Sensor Web Enablement (SWE) standards in marine scenarios
- Improve semantic interoperability through the use of vocabulary services
- Provide templates for the description of observatories/sensor devices and for encoding their observation data

Consequently, this specification of the SeaDataCloud SWE Ingestion Service will rely on the (intermediate) results of the Marine SWE Profiles Working Group.

3.4. Vocabularies

The OGC SWE standards introduced in the previous sub-sections are intended to ensure syntactic interoperability. They define how to invoke certain operations and to encode data as well as metadata. However, they do not deal with semantic aspects such as ensuring a common terminology.

For example, many fields within the SWE specifications are soft-typed. This means that there are generic fields which are assigned a meaning through a definition attribute. But these definitions are not covered by the SWE standards.

Instead there is a need for a common set of terms that can be used as definitions but also property values within SWE compliant documents. Such a common set of terms is ensured through the use of vocabularies. For the SeaDataCloud project, the vocabulary servers of NERC/BODC will be used[3]. This approach is in line with the activities of the SeaDataCloud work package 8, which addresses the advancement of these vocabularies.

Also within the Marine SWE Profiles working group, the NERC vocabulary servers play a significant role as these profiles will provide guidance how these vocabularies can be used to achieve semantic interoperability within marine SWE systems (Kokkinaki et al., 2016).

[3] http://vocab.nerc.ac.uk/

4. Progress on Marine SWE Profiles

In section 3.3 the activities of the Marine SWE Profiles working group were introduced. As these developments are of high importance for the SeaDataCloud project, a significant contribution was made to this group. On the one hand, the Marine SWE Profiles will help to harmonise the usage of the OGC SWE standards across a broad range of institutions and projects. On the other hand, it is of direct relevance for the specification of the SWE Ingestion Service.

The SWE Ingestion Service requires methods for describing the outputs (data structure) delivered by a sensor or a sensor data file as well as an approach for formally defining how to interact with a sensing device (i.e. which commands/parameters are understood). For this purpose the SensorML standard provides an ideal foundation. However, due to multiple ways how this can be achieved using SensorML, there is a need to harmonise this as part of the SWE Marine Profiles. Consequently, the contribution of the SeaDataCloud project especially addressed this question.

The development of the Marine SWE Profiles is supported by a large group of individuals (e.g. there are currently more than 100 members registered on the group mailing list) as well as organisations and projects such as AODN, BRIDGES, envri+, EUROFLEETS/EUROFLEETS2, FixO3, FRAM, IOOS, Jerico/Jerico-Next, NeXOS, ODIP/ODIP II, RITMARE, SeaDataCloud/SeaDataNet, SenseOcean, X-DOMES. On the one hand this large participation helps to ensure broad acceptance of the developed profiles, on the other hand this leads to a more complex discussion process which results in a longer development time for the Marine SWE Profiles than originally expected.

However, despite the mentioned challenges, the Marine SWE Profiles Working Group has already achieved first results. The first activities had a strong focus on collecting usage examples of the SWE specifications in different projects and organisations. These findings are documented in a Wiki hosted by 52°North (see Figure 1). This includes not only examples of O&M and SensorML but also endpoints of available SOS instances.

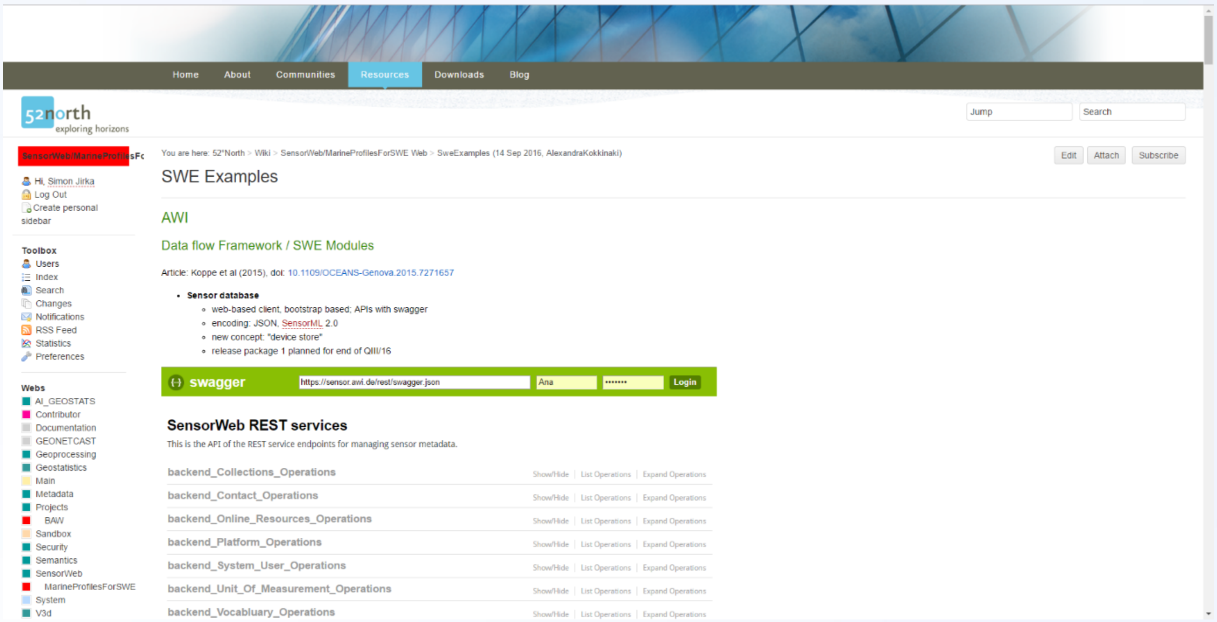

Figure 1: Screenshot of the Marine SWE Profiles Wiki

In addition, a first model how a SensorML profile for marine sensors could be structured was developed. This has resulted in a hierarchical approach that distinguishes between:

- Sensor platforms (e.g. vessels, gliders, buoys)

- Instruments (attached to sensor platforms)
- Detectors (as component of an instrument)

However, depending on the actual organisation, only selected layers of the three levels above might be used. Furthermore, the proposed profile distinguished between the descriptions of types and instances. This means that a manufacturer could provide a description of a sensor type (e.g. through the Esonet/FixO3 Yellow Pages) while a sensor operator would only have to provide the specific information of the sensor instance of this type that is deployed by the operator. Figure 2 shows an exemplary overview of the possible elements that describe an instance of an instrument.

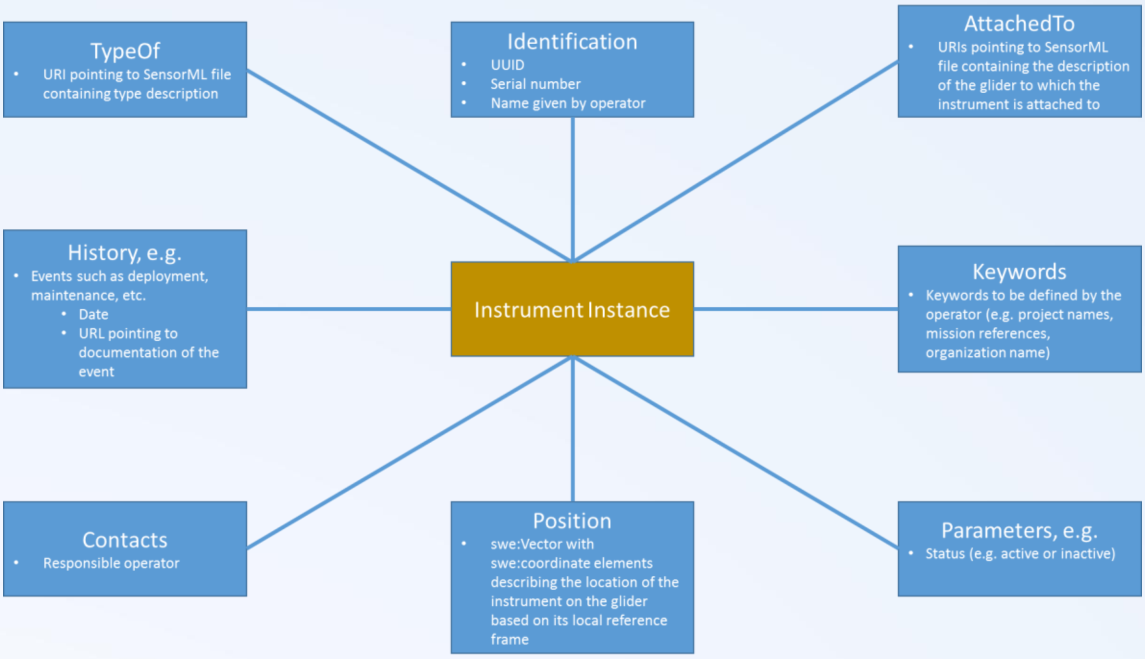

Figure 2: Excerpt from the Ideas for the Marine SensorML Profile Development (here: Description of an Instrument Instance)

Another important result of the Marine SWE Profiles Working Group is the development of an approach how to describe sensor outputs as well as commands within SensorML documents. As this aspect was a potential blocker for the SeaDataCloud SWE Ingestion Service, special emphasize was put on this specific aspect. This has led to first stable results which will serve as an input for the SeaDataCloud developments. The results of this discussion process of the Marine SWE Profiles Working Group are documented in the sections 5 and 6 of this deliverable. In addition, the SeaDataCloud partner BODC has undertaken efforts to work on additional vocabularies needed for semantically enabling the description of sensor inputs and outputs (e.g. terms for specifying the underlying SWE concepts or for classifying sensor commands).

As part of the SeaDataCloud contribution to the Marine SWE Profiles activities, a collaborative approach for documenting the results of the working group has been explored. This approach is based on using Github, a popular platform for hosting open source community projects, so that all interested parties may propose and submit changes to the Marine SWE Profiles. Figure 3 shows a screenshot of the resulting documentation.

Data Access

The following SOS operations are necessary for enabling the access to observation data:

Operation		Remarks
GetCapabilities	Mandatory	As specified in the SOS 2.0 standard
GetObservaton	Mandatory	As specified in the SOS 2.0 standard
DescribeSensor	Mandatory	As specified in the SOS 2.0 standard
GetFeatureOfInterest	Optional	As specified in the SOS 2.0 standard
GetDataAvailability	Optional	Not part of the SOS 2.0 standard; should be implemented as specified in the INSPIRE Technical Guidance Documents on O&M and SOS-based Download Services

GetCapabilities

This operation is the main entry point to every OGC Web Services. A GetCapabilities response provides metadata on an SOS servers' capabilities (e.g. supported operations) and an overview description of it contents. The content of an SOS server is organised as so called "ObservationOfferings" which can be considered as the datasets offered by the SOS server.

The SOS standard allows different ways how to organise the available observation data into ObservationOfferings. Due to the heterogeneity of the marine use cases that may benefit from the SOS interface, it is not possible to provide a common rule how such data sets shall be defined.

Recommendation SOS_01: An SOS server shall provide all marine datasets via separate ObservationOfferings. In addition, the SOS 2.0 standard requires that one ObservationOffering may only contain one procedure.

Typical approaches for structuring ObservationOfferings are: * All measurements taken by a specific sensor * All measurements taken by a sensor during a certain time period * All measurements taken by a sensor network (modelled a single procedure) for an observed property * All measurements taken during a voyage of a research vessel (vessel modelled as a procedure) * ...

DescribeSensor

This operation allows the retrieval of metadata describing the process through which observation data were

Figure 3: Screenshot of an initial Github page to document the Marine SWE Profiles

In addition, there have been several practical implementations of SeaDataCloud partners to evaluate and demonstrate the feasibility of the developed elements of the Marine SWE Profiles. For example, BODC, OGS and Ifremer have set up different SOS implementations as part of the EMODNET project. These different implementations of SOS servers are subsequently integrated into the 52°North Helgoland SOS viewer. For this purpose the 52°North Sensor Web proxy is used that harvests the metadata exposed by these servers and subsequently enables a lightweight visualisation. A fully running version of this set-up is expected by mid-October 2017. As these implementations already use certain agreements of the Marine SWE Profiles Working Group, this activity already illustrates the gained interoperability advantages.

5. SWE Ingestion Architecture and Interfaces

This chapter introduces the developed SWE Ingestion Architecture as well as the corresponding interfaces needed for implementing the SWE Ingestion Service.

5.1. Overview

Figure 4 provides an overview of the SWE Ingestion Architecture. The core of this architecture is an SOS server (due to its central role in the SWE Ingestion Architecture the term SOS is used in this document synonymously to the term "SWE Ingestion Service"). This server is capable of managing data and metadata about a range of different observation data sources. These include:

- Smart sensors (for example sensors supporting the SEISI concepts developed by the NeXOS project (Toma et al., 2014)) that are able to push their metadata and data autonomously to an SOS server.
- Sensor platforms that are regularly queried by an importer for new available data. In this case a description of the sensor platform interface is necessary so that an importer can be automatically configured to the commands and data formats of the sensor platform.
- Automatic harvesting of data published on a server (e.g. CSV files put on an FTP server). In this case a description of the data structure is necessary so that the importer can be configured to the specific format in which the data is provided.
- Uploading (e.g. by researchers) so that the data is automatically processed. Also in this case a description of the data structure is necessary so that the importer can be configured to the specific format in which the data is provided.
- Further data sources that need to be covered in the future (e.g. Internet of Things data sources such as MQTT brokers (see for example the activities of the Marine Institute, Ireland)).

The harvested metadata and data can subsequently be coupled with the SeaDataCloud infrastructure. An important element of this data flow is the provision of the collected information through the SeaDataCloud Common Data Index (CDI) so that researchers are able to discover and access the available data sets. In this case, the SOS server will act as a feeder for the CDI through interfaces offered by the CDI.

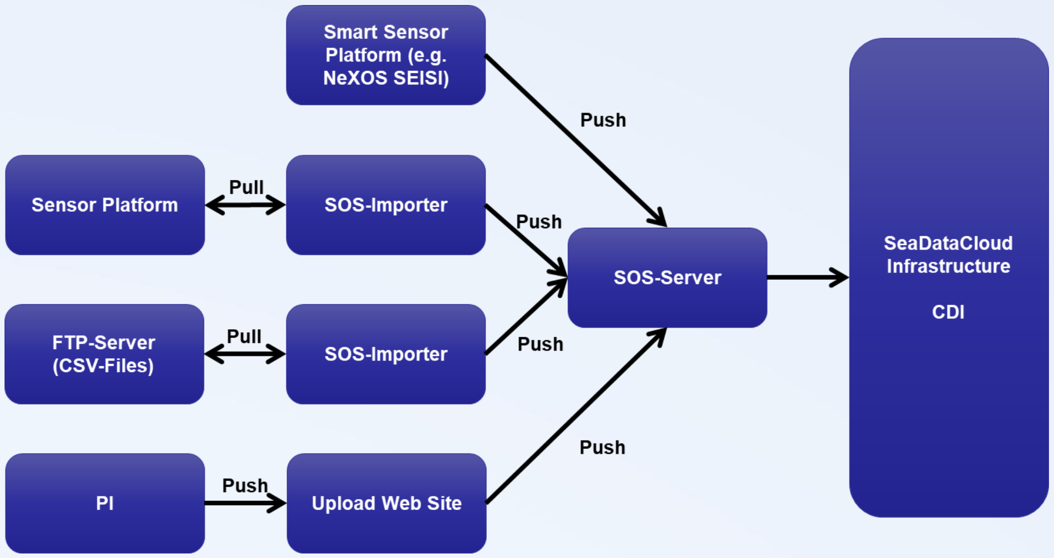

Figure 4: SWE Ingestion Architecture

The next sections will introduce several building block that can be used for realising different types of data publication workflows for the previously mentioned data sources.

5.2. Ingestions Workflows

For uploading metadata and observation data to an SOS server, two workflows will be offered that are based on the OGC Sensor Observation Service (SOS) 2.0 standard. They can be used for covering the following links within the SWE Ingestion Architecture:

- Pushing of data by a smart sensor to an SOS server
- Pushing of previously collected data from an SOS importer to an SOS server
- Pushing of data that was inserted through a data upload Web site

Figure 5 shows the first option, the use of the transactional operations offered by the SOS 2.0 interface. In this case a sensor is first registered at an SOS server (i.e. by providing the sensor metadata) and subsequently new data is inserted through a series of InsertObservation requests. The InsertObservation requests can be executed regularly every time new data is available. This approach is rather easy to handle as it involves only two types of SOS operation call, however through the use of the InsertObservation operation, there is usually a higher overhead through the XML representation of the data that shall be inserted.

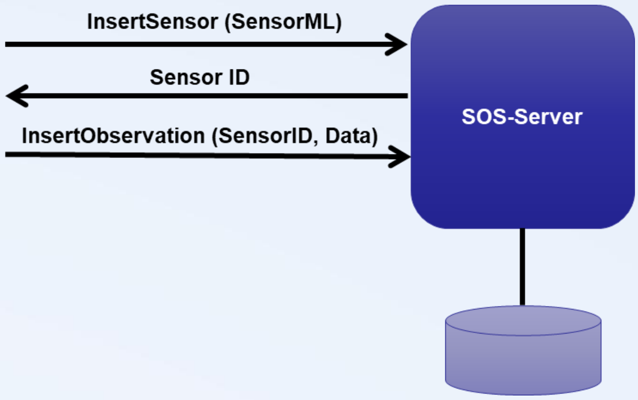

Figure 5: Pushing of Data through the Transactional SOS Operations

The alternative workflow is illustrated in Figure 6. This is based on the result handling operations of the SOS 2.0 interface. Also in this case the sensor is registered trough the InsertSensor operation. However, after this, a different approach is followed. Instead of always inserting complete observations (in XML representation), the idea is to avoid the repetitive transmission of data that is common to all observations (e.g. the properties observed within a data set often remain constant, thus, there is no need to repeat the identifiers of the observed properties). To achieve this, the publisher of the observation data, sends first a template (InsertResultTemplate) that describes the structure of the data that will be inserted (e.g. which fields are contained in the data). Furthermore, this template contains all information which is static (e.g. the measurement location (feature of interest) in case of a stationary, fixed sensor). After that, the publisher can send the data to the SOS in a compact format (basically using a payload of comma-separated values) through the InsertResult operation. This leads to a slightly higher complexity because the result templates need to be managed and interpreted. However, at the same time the volume of the transmitted data can be significantly reduced. Thus, this approach is especially advantageous in case of resource constrained communication links.

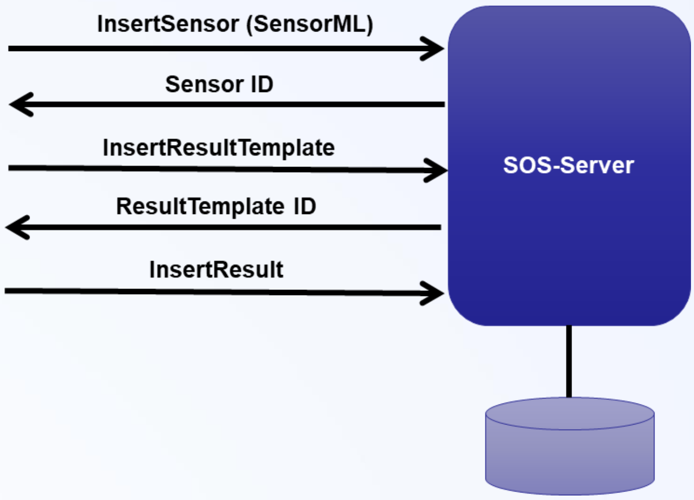

Figure 6: Pushing of Data through the SOS Result Handling Operations

The following two sub-sections describe these two workflows in a more formal manner through UML sequence diagrams.

5.2.1. Data Ingestion Workflow based on the Transactional SOS Operations

As shown in Figure 7, the first step of this workflow is the registration of a sensor at the SWE Ingestion Service through the InsertSensor request. The payload of this request shall be a SensorML document as defined in section 5.3. The response to the InsertSensor operation is an XML document as specified by the OGC SOS 2.0 standard that contains the identifier which has been assigned to the sensor by the SWE Ingestion Service.

Subsequently the data publisher is able to send data to the SWE ingestion server as described in section 5.4.1. During these calls, the publisher refers to the previously assigned sensor identifier (InsertSensor response) and provides the data that shall be uploaded.

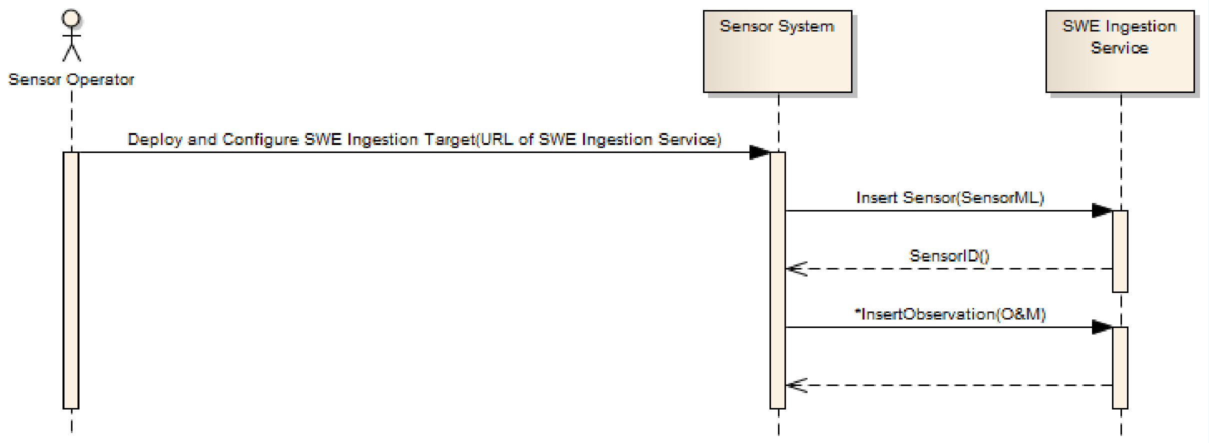

Figure 7: Data Ingestion Workflow based on the Transactional SOS Operations

5.2.2. Data Ingestion Workflow based on the SOS Result Handling Operations

Figure 8 illustrates this alternative workflow. As in the previous workflow, the first step is the registration of the sensor at the SWE Ingestion Service through the InsertSensor operation. The payload of this request shall be a SensorML document as defined in section 5.3. The response to the InsertSensor operation is an XML document as specified by the OGC SOS 2.0 standard that contains the identifier which has been assigned to the sensor by the SWE Ingestion Service.

After this, a template describing the data of the sensor that has been registered is transmitted to the SWE Ingestion Service (see section 5.4.2.1). Subsequently, InsertResult requests can be send to the SWE Ingestion Service in order to upload data that complies with the previously inserted template (see section 5.4.2.2).

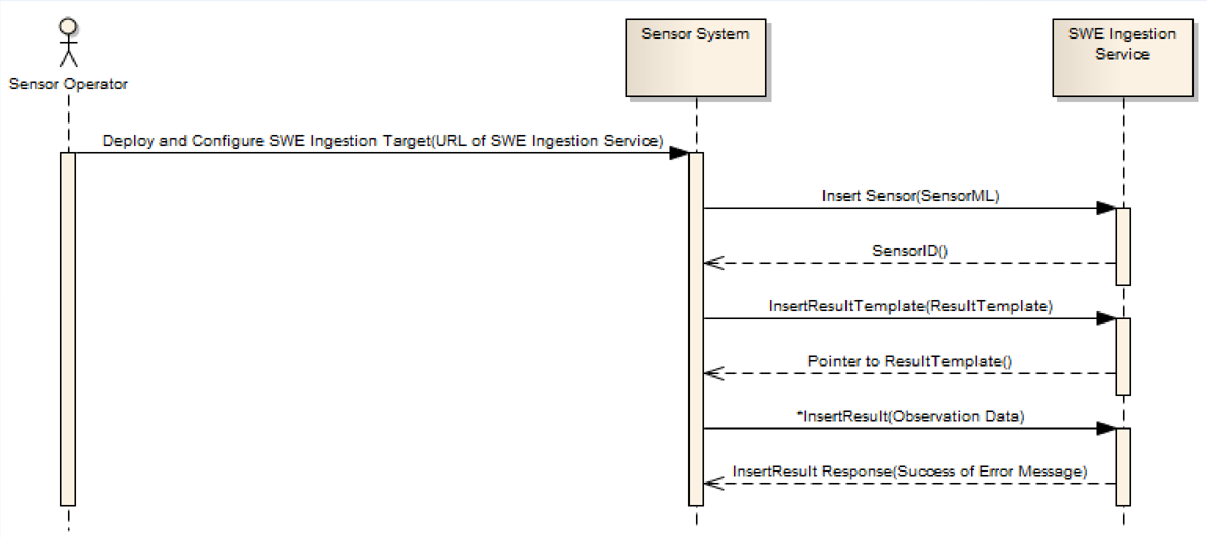

Figure 8: Data Ingestion Workflow based on the SOS Result Handling Operations

5.3. Sensor Registration

When registering a new sensor, it is necessary to provide the metadata of this sensor as payload of an InsertSensor request as specified by the OGC SOS 2.0 standard. The payload of this InsertSensor request comprises especially a SensorML 2.0 encoded sensor description. Because the SensorML standard intentionally offers a high degree of freedom, this specification provides further guidance (a profile) which information shall be at least included in a SensorML document submitted to the SeaDataCloud SWE Ingestion Service. Table 1 provides a detailed overview of the required SensorML elements (a Schematron-based definition of the corresponding SensorML Profile is available in Annex I):

Table 1: Overview of SensorML Elements needed for providing Sensor Descriptions

Name	SensorML Element	Description
Keywords	/sml:keywords/sml:KeywordList/ sml:keyword	Human readable terms as well as references to applicable vocabulary entries that describe the sensor/data stream (see SeaDataCloud WP8)
Identification	/sml:identification/sml:IdentifierList/ sml:identifier	Identifiers that can be used for referring to the specific sensor. Each identifier element shall have a term element with a definition attribute that refers to a vocabulary entry (see SeaDataCloud WP8) which explains the meaning of the identifier. There shall at least be one unique identifier.

Classification	/sml:classification/sml:ClassifierList/ sml:classifier	(Only if applicable) Classifiers describing the nature/type of sensor/data stream. For each classifier there shall be a definition and a value that both refer to a corresponding vocabulary entry (see SeaDataCloud WP8).
Characteristics	/sml:characteristics/sml:CharacteristicList/ sml:characteristic	(Only if applicable) Descriptions of important properties of a sensor that are relevant for interpreting the delivered data stream. For each characteristic there shall be a definition attribute that refers to a vocabulary entry (see SeaDataCloud WP8) which explains the meaning of the characteristic.
Capabilities	/sml:capabilities/sml:CapabilityList/ sml:capability	(Only if applicable) Descriptions of important capabilities (e.g. precision) of a sensor that are relevant for interpreting the delivered data stream. For each capability there shall be a definition attribute that refers to a vocabulary entry (see SeaDataCloud WP8) which explains the meaning of the capability.
Contacts	/sml:contacts/sml:ContactList/sml:contact	Relevant contacts for the sensor/data stream. There shall at least be one contact element that provides contact information about the principal investigator (role code: CI_RoleCode_ principalInvestigator, code list: http://www.isotc211.org/ 2005/resources/Codelist/ gmxCodelists.xml#CI_RoleCode/)

Outputs	/sml:outputs/sml:OutputList/sml:output	Description of the data stream delivered by the sensor. For each field in the data stream, there shall be a dedicated output element as defined by the OGC SensorML 2.0 standard. Each output shall contain a definition attribute that refers to a vocabulary entry (see SeaDataCloud WP8) which explains the meaning of the output (e.g. time stamps, names of observed properties, etc.).
Position	/sml:position	(Only for stationary sensors) Location of the sensor.

5.4. Upload

5.4.1. Transactional SOS Operations

After a sensor has been registered as described in the previous section, it is possible to insert data measured by this sensor through the InsertSensor operation. The InsertSensor requests shall follow the structure as shown in the annotated template below:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertObservation>
   <!--The offering (~data set) to which the data shall be added. This identifier
       has been obtained in the initially executed InsertSensor call.-->
   <sos:offering>http://www.52north.org/test/offering/7846272</sos:offering>
   <sos:observation>
      <om:OM_Observation gml:id="o1">
         <!--The phenomenon time indicates the point or period in time to which the
             measured value applies.-->
         <om:phenomenonTime>
            <gml:TimeInstant gml:id="phenomenonTime">
               <gml:timePosition>2017-08-19T13:30:00+00:00</gml:timePosition>
            </gml:TimeInstant>
         </om:phenomenonTime>
         <!--The result time can be used for expressing the time when the
             observation was made available. As the time of publication and
             measurement are usually treated as equal, the result time can consist
             of a reference to the phenomenon time.-->
         <om:resultTime xlink:href="#phenomenonTime"/>
         <!--The sensor which has taken the measurement. This identifier has been
             obtained in the initially executed InsertSensor call.-->
         <om:procedure xlink:href="http://www.52north.org/test/procedure/
                                   44567575"/>
         <!--The parameter which was observed. This shall be provided as a
             reference to a dictionary entry, in which the observed property is
             defined (see SeaDataCloud WP8).-->
         <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/collection/
                                          P01/current/CNDCZZ01/"/>
         <!--The feature of interest refers to the geospatial object (e.g.
             location) for which the sensor is measuring data. For a stationary
```

```
            sensor this is usually a point. For a mobile sensor this may refer to
            the bounding box in which the sensor is moving.-->
        <om:featureOfInterest>
            <sams:SF_SpatialSamplingFeature gml:id="5247574778">
                <!--The identifier of the feature.-->
                <gml:identifier codeSpace="">
                    http://www.52north.org/test/featureOfInterest/95536
                </gml:identifier>
                <!--A human readable name of the feature location.-->
                <gml:name>52°North location</gml:name>
                <!--The type of the feature.-->
                <sf:type xlink:href="http://www.opengis.net/def/
                                    samplingFeatureType/OGC-
                                    OM/2.0/SF_SamplingPoint"/>
                <!--The geometry of the feature encoded as defined by the OGC GML
                    standard.-->
                <sams:shape>
                    <gml:Point gml:id="test_feature_9">
                        <gml:pos srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
                            51.935101100104916 7.651968812254194
                        </gml:pos>
                    </gml:Point>
                </sams:shape>
            </sams:SF_SpatialSamplingFeature>
        </om:featureOfInterest>
        <!--The value that was measured.-->
        <om:result xsi:type="swe:Quantity">
            <swe:Quantity definition="http://vocab.nerc.ac.uk/collection/
                                    P01/current/CNDCZZ01">
                <!--The unit of measurement.-->
                <swe:uom code="SIE"/>
                <!--The value.-->
                <swe:value>5</swe:value>
            </swe:Quantity>
        </om:result>
    </om:OM_Observation>
  </sos:observation>
</sos:InsertObservation>
```

5.4.2. SOS Result Handling Operations

A more compact alterative to the InsertObservation operation are the Result Handling operations of the SOS interface. In this case, two types of requests are necessary. First, a template describing the structure of the data as well as properties that are common to all observations, is inserted through the InsertResultTemplate request. After that, data can be inserted through the InsertResult request by following the previously registered template.

5.4.2.1. Data Template Registration (InsertResultTemplate)

The insertion of result templates shall be performed by following the annotated template shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertResultTemplate>
   <sos:proposedTemplate>
      <sos:ResultTemplate>
      <!--The identifier of the result template. Please note: For the InsertResult
          operation this identifier is not used. Instead the identifier returned in
          the response to this request shall be used (see OGC SOS 2.0 standard).-->
         <swes:identifier>64858d4e-cb95-4f0e</swes:identifier>
```

```xml
<!--The offering (~data set) to which the data shall be added. This identifier
    has been obtained in the initially executed InsertSensor call.-->
<sos:offering>http://www.52north.org/test/offering/7846272</sos:offering>
      <sos:observationTemplate>
         <om:OM_Observation gml:id="sensor2obsTemplate">
            <om:type xlink:href="..."/>
            <!--The fields for phenomenon time and result time remain empty as
                this is just a result template.-->
            <om:phenomenonTime nilReason="template"/>
            <om:resultTime nilReason="template"/>
            <!--The sensor which has taken the measurement. This identifier has
                been obtained in the initially executed InsertSensor call.-->
            <om:procedure xlink:href="http://www.52north.org/test/procedure/
                                      44567575"/>
            <!--The parameter which was observed. This shall be provided as a
                reference to a dictionary entry, in which the observed property
                is defined (see SeaDataCloud WP8).-->
            <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/collection/
                                             P01/current/CNDCZZ01"/>
            <!--The feature of interest refers to the geospatial object (e.g.
                location) for which the sensor is measuring data. For a
                stationary sensor this is usually a point. For a mobile sensor
                this may refer to the bounding box in which the sensor is
                moving.-->
            <om:featureOfInterest>
               <sams:SF_SpatialSamplingFeature gml:id="5247574778">
                  <!--The identifier of the feature.-->
                  <gml:identifier codeSpace="">
                     http://www.52north.org/test/featureOfInterest/95536
                  </gml:identifier>
                  <!--A human readable name of the feature.-->
                  <gml:name>52°North location</gml:name>
                  <!--The type of the feature.-->
                  <sf:type xlink:href="http://www.opengis.net/def/
                                       samplingFeatureType/OGC-OM/2.0/
                                       SF_SamplingPoint"/>
                  <!--The geometry of the feature encoded as defined by the OGC
                      GML standard.-->
                  <sams:shape>
                     <gml:Point gml:id="test_feature_9">
                        <gml:pos srsName="http://www.opengis.net/def/crs/
                                          EPSG/0/4326">
                           51.935101100104916 7.651968812254194
                        </gml:pos>
                     </gml:Point>
                  </sams:shape>
               </sams:SF_SpatialSamplingFeature>
            </om:featureOfInterest>
            <!--The result field remains empty as this is just a result
                template.-->
            <om:result/>
         </om:OM_Observation>
      </sos:observationTemplate>
      <sos:resultStructure>
         <swe:DataRecord>
            <!--The phenomenon time field shall be indicated as shown here:-->
            <swe:field name="phenomenonTime">
               <swe:Time definition="http://www.opengis.net/def/property/OGC/
                                     0/PhenomenonTime">
                  <swe:uom xlink:href="http://www.opengis.net/def/uom/
                                       ISO-8601/0/Gregorian"/>
               </swe:Time>
            </swe:field>
            <!--The fields containing the observed values shall be described as
                shown below. For each data field a corresponding field element
                shall be added.-->
```

```
                <swe:field name="sea_water_conductivity">
                   <!--The parameter which was observed. This shall be provided as a
                       reference to a dictionary entry, in which the observed
                       property is defined (see SeaDataCloud WP8).-->
                   <swe:Quantity definition="http://vocab.nerc.ac.uk/collection/
                                            P01/current/CNDCZZ01">
                      <!--The unit of measurement.-->
                      <swe:uom code="SIE"/>
                   </swe:Quantity>
                </swe:field>
             </swe:DataRecord>
          </sos:resultStructure>
          <!--The result encoding element describes how the values transmitted in an
              InsertResult request are encoded. While the token separator marks the
              fields within an observation, the block separator allows to mark
              observations.-->
          <sos:resultEncoding>
             <swe:TextEncoding tokenSeparator="#" blockSeparator="@"/>
          </sos:resultEncoding>
       </sos:ResultTemplate>
   </sos:proposedTemplate>
</sos:InsertResultTemplate>
```

5.4.2.2. Data Upload (InsertResult)

For the data insertion through the InsertResult operation the approach as explained in the annotated example below shall be used:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertResult>
   <!--Reference to the previously registered result template.-->
   <sos:template>
      64858d4e-cb95-4f0e
   </sos:template>
   <!--The data encoded following the structure in the result template. In this
       example, there are three data blocks consisting of two fields.-->
   <sos:resultValues>
      3@2017-04-19T13:30:00+02:00#5.4@2017-0419T13:31:00+02:00#5.1
       @2017-04-19T13:33:00+02:00#5.3
   </sos:resultValues>
</sos:InsertResult>
```

6. SOS Importer

The previous section has covered the transfer of metadata and data into a repository relying on the transactional or result handling operations defined by the OGC SOS standard. However, if data shall be autonomously be pulled from a sensor or a set of sensor data files on a server, further logic is necessary to describe sensor commands as well as the structure of delivered data sets/streams.

In the following, section 6.1 specifies the process for directly collecting observation data from sensing devices, while section 6.2 addresses the collection and automated parsing of data from file-based archives.

6.1. Bridging between Sensors and the SWE Ingestion Service

For allowing an automatic bridging between sensors and the SWE Ingestion Service two aspects are important. On the one hand, it is necessary to define how the SOS Importer can retrieve the measured data from a sensing device. This concerns especially the commands that have to be sent to the sensor (section 6.1.1). After data has been requested from a sensing device, the next step is to decode this data stream. For that purpose a formal definition of the structure of the data is needed (see section 6.1.2).

For both of these aspects, the SeaDataCloud SWE Ingestion Service relies on the OGC SensorML standard. The following two subsections provide a corresponding profile of the necessary SensorML sections which are relevant to fulfil these requirements. In both cases dedicated section of non-physical SensorML components (e.g. a non-physical component could be a command that can be executed by the sensor) are used. This means that each command with its input parameters and outputs is described as a dedicated non-physical SensorML component.

6.1.1. Describing Sensor Commands

For describing commands that can be executed by a sensing device (e.g. asking the sensor to return (a subset of) measured data), the Inputs section of a non-physical SensorML component shall be used. The corresponding outputs shall be described as an outputs section within the same SensorML document (see section 6.1.2). A Schematron-based definition of the corresponding SensorML Profile is available in Annex I).

Table 2: Overview of SensorML Elements needed for Describing Sensor Commands

Name	SensorML Element	Description
Identifier	/sml:SimpleProcess/sml:identification/ sml:IdentifierList/sml:identifier	There shall be at least one unique identifier for referring to the command. The term element within the identifier shall have a definition attribute which refers to a vocabulary entry defining the meaning of the identifier (see SeaDataCloud WP8).

Classifier	/sml:SimpleProcess/sml:classification/sml:ClassifierList/sml:classifier	There shall be at least one classifier describing the nature of the command (e.g. data access, configuration change, etc.). The term defining the classifier shall have a definition attribute which refers to a vocabulary entry defining the meaning of the classifier. Also the value of the classifier shall refer to a corresponding vocabulary entry (see SeaDataCloud WP8).
Input Parameters	/sml:SimpleProcess/sml:inputs/sml:InputList/sml:input/sml:DataInterface/sml:data/swe:DataStream/swe:elementType/swe:DataRecord/swe:field	For each input parameter of the command there shall be a dedicated field in the inputs section. Each field shall be structured as defined by the OGC SensorML 2.0 standard. Each field shall provide at least information about the data type of the input, the definition of the input parameter (referring to a vocabulary entry; see SeaDataCloud WP8), a label and if applicable the unit of measurement.
Encoding	/sml:SimpleProcess/sml:inputs/sml:InputList/sml:input/sml:DataInterface/sml:data/swe:DataStream/swe:encoding	Within the inputs section a SensorML 2.0 compliant encoding element shall be provided that defines how the input parameters shall be encoded when they are transmitted to the sensor.

6.1.2. Describing Sensor Outputs

While the previous section defines the way the input parameters of sensor commands shall be described, this section specifies how the definition of sensor outputs shall be provided. In this case, the outputs section of a non-physical SensorML process shall be used as defined in (a Schematron-based definition of the corresponding SensorML Profile is available in Annex I).

Table 3 (a Schematron-based definition of the corresponding SensorML Profile is available in Annex I).

Table 3: Overview of SensorML Elements needed for Describing the Structure of Observation Data Sets and Streams

Name	SensorML Element	Description

Output fields	/sml:SimpleProcess/sml:outputs/sml:OutputList/ sml:output/sml:DataInterface/sml:data/ swe:DataStream/swe:elementType/ swe:DataRecord/swe:field	For each output field of the command (e.g. time stamps, measured values, flags, etc.) there shall be a dedicated field in the outputs section. Each field shall be structured as defined by the OGC SensorML 2.0 standard. Each field shall provide at least information about the data type of the output, the definition of the output field (referring to a vocabulary entry; see SeaDataCloud WP8), a label and if applicable the unit of measurement.
Encoding	/sml:SimpleProcess/sml:outputs/sml:OutputList/ sml:output/sml:DataInterface/sml:data/ swe:DataStream/swe:encoding	Within the outputs section a SensorML 2.0 compliant encoding element shall be provided that defines how the output data is encoded when returned by the sensor.

6.2. Loading Data Files to the SWE Ingestion Service

If data is available as files which are collected in a certain location, the SOS importer shall be able to automatically interpret the structure of these files. Especially the following aspects need to be described:

- Fields contained in the data
    - Data types
    - Meaning (e.g. properties that are observed, time stamps)
- Common properties for all delivered data, e.g.
    - Reference to sensor/platform
    - Feature of interest
    - Observed properties
    - Units of measurement

For this purpose the user interfaces described in section 7.1 shall be available for data providers. This editor shall produce a machine-readable data structure (a SensorML Output) description that follows the approach as described in section 6.1.2.

7. User Interface Requirements

Besides the specification of the interactions between the different components of the SWE Ingestion Architecture, there are also further requirements that must be considered during the development of the corresponding implementations. This concerns especially the interfaces that are provided to users for interacting and using the SWE Ingestion workflow.

Consequently the following subsections describe different aspects of user interactions with the SWE Ingestion mechanisms. This comprises especially an editor for describing the structure of incoming data sets/data streams as well as sensor interfaces and corresponding metadata (section 7.1). Furthermore, a simple user interface for retrieving the published metadata (section 7.2) and a feedback mechanism to data providers (section 7.3) are needed.

7.1. Description of Data Structures and Sensor Interfaces as well as Provision of Metadata

As specified in sections 4 and 6, the description of sensors as well as sensor outputs (data streams) shall be based on the OGC Sensor Model Language (SensorML) standard. While the SensorML XML encoding is commonly used for machine-to-machine communication, it cannot be expected that researchers working with marine sensors manually edit these rather complex files. Instead a graphical user interface is needed for comfortably entering the necessary information.

A starting point for this development will be the 52°North smle implementation, a Web-based editor for OGC compliant SensorML 2.0 files[4] (see Figure 9). This development has been performed in a first version as part of the European FP7 projects NeXOS[5] and FixO3[6]. As this editor is available under an open source license, SeaDataCloud will be able to build on these previous results and extend it with the necessary functionality.

Within SeaDataCloud, especially the following additional functionality will be added to smle:

- Description of sensor/platform commands (e.g. for downloading measured data): This requires and additional functionality to create non-physical processes with inputs/parameters and corresponding outputs.
- Description of sensor outputs by providing information about the fields of a sensor data stream such as data types, units of measurement, observed properties, time stamps, geographic information and syntactic information (block separators, token separators, decimal separators, etc.).
- Support of vocabularies: To the extend supported by the NERC vocabulary servers, smle shall support the discovery and selection of terms from existing vocabularies so that users are able to define the meaning of a (soft-typed) field as well as select appropriate terms to fill in textual inputs fields (e.g. keywords and classifiers).

These enhancement shall follow the data specifications defined in sections 4 and 6.

[4] https://github.com/52North/smle

[5] http://www.nexosproject.eu/

[6] http://www.fixo3.eu/

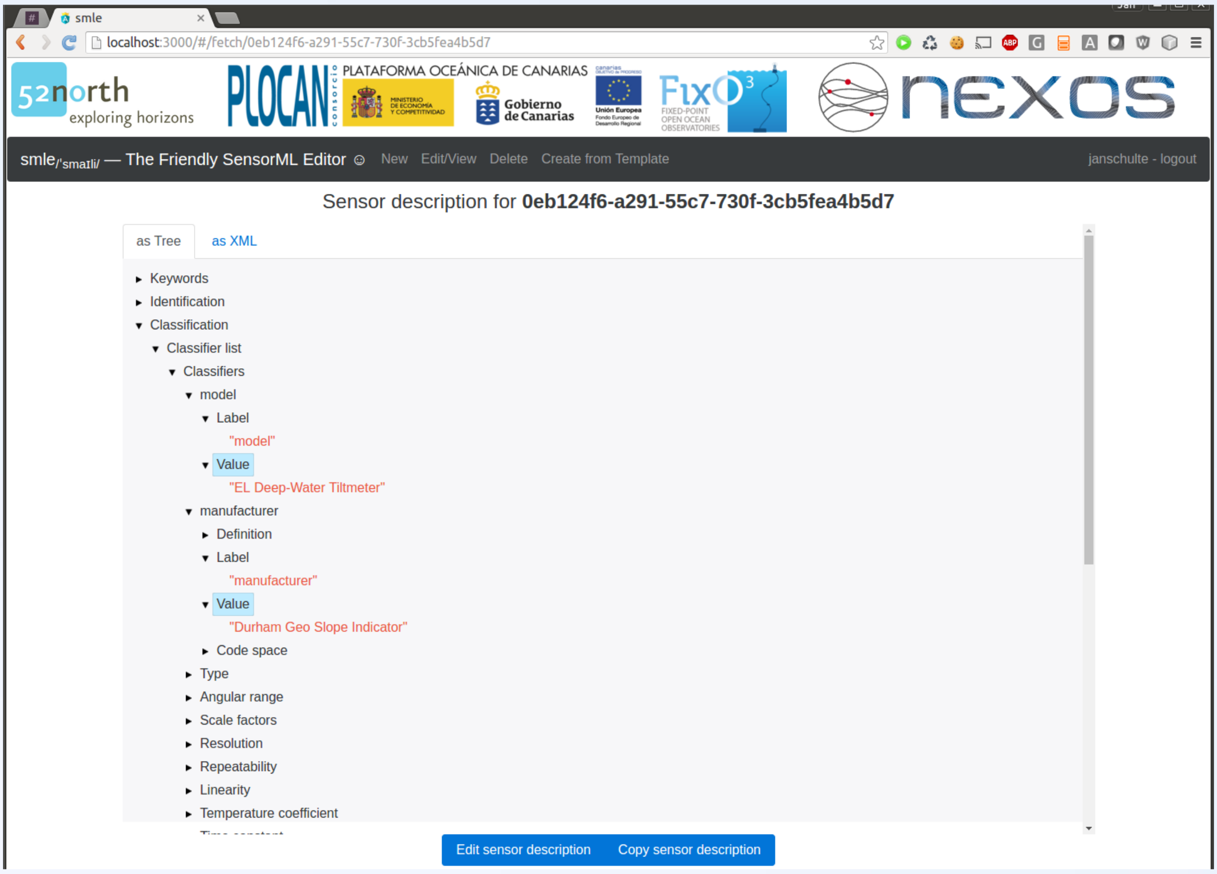

Figure 9: Screenshot of smle

7.2. Retrieval of Sensor Metadata

While the CDI already provides comprehensive discovery functionality for published metadata about marine data, the SWE Ingestion Service shall provide a lightweight interface to retrieve the collected sensor metadata. This shall comprise at least the following functionality:

- Query the metadata of a specific sensor
- Full-text query across the stored SensorML metadata documents

This shall be complemented by a lightweight Web-based user interface to query the metadata collected by the SWE Ingestion Service. This functionality shall be implemented as an extension of the smle SensorML editor and in the SeaDataCloud SOS viewing services (D10.18) which will be based on the 52°North Helgoland implementation.

More sophisticated discovery functionality will be provided through the CDI.

7.3. Feedback to Data Providers

In order to improve the quality of the data delivered via the SWE Ingestion Service, it is important that data providers and sensor operators receive useful feedback about the quality and success of the data they have provided. Thus, the SWE Ingestion Service shall provide additional feedback that goes beyond the functionality defined by the OGC SOS standard.

Especially the following information during the data ingestion process shall be logged and made accessible to the data providers in the form of tabular/textual overviews:

- Amount of successfully inserted sensors (metadata)
- Amount of unsuccessfully inserted sensors (metadata); this shall include a list of the failed sensors and the specific reasons for failure
- Amount of successfully inserted observations
- Amount of unsuccessfully inserted observations; this shall include a list of the failed observations and the specific reasons for failure

This information shall be made accessible through a dedicated operation which shall offer at least the following query parameters:

- Identifier of the data provider for which the log file shall contain information
- Time span for which the log file shall be accessed

Besides this, also basic quality check should be investigated. Depending on resource availability, the following check will be considered: SensorML allows to provide the value range of a sensor/observed property. This could be used by the SWE Ingestion Service to check if a delivered observation result is within the range of valid measurements.

8. Conclusions and Outlook

This deliverable has introduced the specification of the SeaDataCloud SWE Ingestion Service. It is strongly based on interoperability standards for handling measurement data (especially the OGC Sensor Web Enablement Standards including OGC SOS, O&M as well as SensorML).

This component is expected to facilitate the publication of (real-time) data streams within the SeaDataCloud infrastructure by enabling plug-and-play mechanisms for sensor integration and interpretation of observation data streams/data sets. Thus, by reducing the necessary efforts to publish data, potentially directly as an output of a sensor platform or a simple file based data store, the efforts that need to be invested by data providers are reduced. In combination with dissemination efforts in a later stage of the project, there is a hope for motivating further data providers to publish their (real-time) data via the SeaDataCloud infrastructure, increasing the amount of data that is available to scientists.

With the completion of this initial specification document, the development of the SeaDataCloud SWE Ingestion Service will be kicked-off. Following an iterative and agile development process, the experiences gained during the development and the evaluation of intermediate versions of the SWE Ingestion Service will be incorporated into updates of this specification. Although not published as an official deliverable, the public will benefit from the gathered experiences through an additional, enhanced version of the specification. In addition, several ideas and suggestions were noted as potential future work that might further extend the approach described in this deliverable. These future work items comprise:

- JSON/JSON-LD-based data access endpoints are becoming more and more popular. Thus, as part of future work, also JSON/JSON-LD-based data flows should be supported. For this purpose, the Ingestion Service will be designed in a modular manner so that corresponding connectors can be added at a later point in time.
- The current approach of restricting the existing SensorML standard through additional rules is rather complex. In the future it should be evaluated if the definition of a stripped-down standalone schema for the data ingestion flows might be better to handle.
- The W3C Semantic Sensor Network (SSN) Ontology offers a semantically rich approach for offering sensor descriptions. As part of future work the SSN should be evaluated as an alternative approach for providing sensor descriptions.

As soon as a first sufficient set of metadata is available, the next step will be the integration of the collected data with the SeaDataCloud infrastructure, especially the Common Data Index (CDI).

Besides this, there are new emerging technologies which need to be considered as extensions of the SWE Ingestion Service beyond the scope of the SeaDataCloud project. For example, there are several protocols from the Internet of Things community, that promise an efficient and light-weight integration of sensing hardware. Examples are MQTT brokers (as currently investigated by the ODIP II[7] project) or the context broker resulting from the European FIWARE activities.

In summary, the SWE Ingestion Service provides a powerful means for integrating observation data streams/sources into the SeaDataCloud infrastructure. Relying on open interoperability standards and an agile development process, it is expected, that the SWE Ingestion Service will help to facilitate the publication of (real-time) observation data and thus increases the availability of marine observation data.

[7] http://www.odip.eu/

9. References

Botts, Mike and Alexandre Robin (2014). OGC Implementation Specification: Sensor Model Language (SensorML) 2.0.0 (12-000). Wayland, MA, USA, Open Geospatial Consortium Inc.

Bröring, Arne, Christoph Stasch and Johannes Echterhoff (2012). OGC Implementation Specification: Sensor Observation Service (SOS) 2.0 (12-006). Wayland, MA, USA, Open Geospatial Consortium Inc.

Cox, Simon (2011). OGC Implementation Specification: Observations and Measurements (O&M) - XML Implementation 2.0 (10-025r1). Wayland, MA, USA, Open Geospatial Consortium Inc.

Delory, Eric and Simon Jirka (2016). A SensorML-based Metadata Model and Registry for Ocean Observatories - A Contribution from the European Projects NeXOS and FixO3. 2016 Ocean Sciences Meeting. New Orleans, LA, USA, American Geophysical Union.

European Parliament and European Council (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). European Union. Strasbourg, France.

INSPIRE MIG sub-group MIWP-7a (2016a). Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE - Version 3.0. Ispra, Italy, INSPIRE Maintenance and Implementation Group (MIG).

INSPIRE MIG sub-group MIWP-7a (2016b). Technical Guidance for implementing download services using the OGC Sensor Observation Service and ISO 19143 Filter Encoding - Version 1.0. Ispra, Italy, INSPIRE Maintenance and Implementation Group (MIG).

ISO TC 211 (2011). ISO 19156:2011 - Geographic information -- Observations and measurements - International Standard. Geneva, Switzerland, International Organization for Standardization.

Jirka, Simon, Daniel Mihai Toma, Joaquin del Rio and Eric Delory (2014). A Sensor Web architecture for sharing oceanographic sensor data. Sensor Systems for a Changing Ocean (SSCO) 2014 at the Sea Tech Week 2014, Brest, France, IEEE.

Kokkinaki, Alexandra, Louise Darroch, Justin Buck and Simon Jirka (2016). Semantically Enhancing SensorML with Controlled Vocabularies in the Marine Domain. GSW 2016 - Geospatial Sensor Webs Conference 2016, Münster, Germany, CEUR-WS.org.

Toma, Daniel Mihai, Joaquin del Rio, Simon Jirka, Eric Delory and Jay Pearlman (2014). Smart electronic interface for Web Enabled Ocean Sensor Systems. Sensor Systems for a Changing Ocean (SSCO) 2014 at the Sea Tech Week 2014, Brest, France, IEEE.

Toma, Daniel M., Joaquín del Rio, Enoc Martínez, Simon Jirka, Eric Delory, Jay Pearlman and Christoph Waldmann (2016). Applying OGC Sensor Web Enablement to Ocean Observing Systems. GSW 2016 - Geospatial Sensor Webs Conference 2016, Münster, Germany, CEUR-WS.org.

Annex I: Schematron Rules

Rules for Sensor Registration

```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
            xmlns:sml="http://www.opengis.net/sensorml/2.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:swe="http://www.opengis.net/swe/2.0"
            xmlns:gml="http://www.opengis.net/gml"
            xmlns:xlink="http://www.w3.org/1999/xlink"
            xsi:schemaLocation="http://www.opengis.net/sensorml/2.0
                                http://schemas.opengis.net/sensorML/2.0/
                                sensorML.xsd"
            schemaVersion="ISO19757-3">
   <ns prefix="sml" uri="http://www.opengis.net/sensorml/2.0"/>
   <ns prefix="gml" uri="http://www.opengis.net/gml"/>
   <ns prefix="swe" uri="http://www.opengis.net/swe/2.0"/>
   <ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
   <ns prefix="sch" uri="http://purl.oclc.org/dsdl/schematron"/>
   <sch:pattern id="SystemValidation">
      <!--There shall be at least one keyword describing the sensor.-->
      <sch:rule context="//sml:PhysicalSystem">
         <sch:assert test="count(sml:keywords/sml:KeywordList/sml:keyword) >= 1">
             Error: There shall be at least one keyword describing the sensor.
         </sch:assert>
      </sch:rule>
      <!--The SensorML document shall provide a unique identifier for the
          sensor.-->
      <sch:rule context="//sml:PhysicalSystem">
         <sch:assert test="count(sml:identification/sml:IdentifierList/*
                           sml:identifier) >= 1">
            There shall at least be one identifier for the sensor.
         </sch:assert>
      </sch:rule>
      <!--For each identifier, there shall be a definition attribute describing the
          meaning of the identifier.-->
      <sch:rule context="//sml:PhysicalSystem/sml:identification/
                         sml:IdentifierList/sml:identifier/sml:Term">
         <sch:assert test="@definition">
            For each identifier, thereshall be a definition attribute describing
            the meaning of the identifier.
         </sch:assert>
      </sch:rule>
      <!--For each classifier, there shall be a definition attribute describing the
          meaning of the classifier.-->
      <sch:rule context="//sml:PhysicalComponent/sml:classification/
                         sml:ClassifierList/sml:classifier/sml:Term">
         <sch:assert test="@definition">
            For each classifier, there shall be a definition attribute describing
            the meaning of the classifier.
         </sch:assert>
      </sch:rule>
      <!--For each characteristic, there shall be a definition attribute describing
          the meaning of the characteristic.-->
      <sch:rule context="//sml:PhysicalComponent/sml:characteristics/
                         sml:CharacteristicList/sml:characteristic/*">
         <sch:assert test="@definition">
            For each characteristic, there must be a definition attribute
            describing the meaning of the characteristic.
         </sch:assert>
      </sch:rule>
```

```
    <!--For each capability element of a sensor, there must be a definition
        attribute describing the meaning of the capability.-->
    <sch:rule context="//sml:PhysicalComponent/sml:capabilities/
                       sml:CapabilityList/sml:capability/*">
       <sch:assert test="@definition">
          For each capability element of a sensor, there must be a definition
          attribute describing the meaning of the capability.
       </sch:assert>
    </sch:rule>
    <!--There shall at least be one contact element that provides contact
        information about the principal investigator (role code:
        CI_RoleCode_principalInvestigator, code list:
        http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#
        CI_RoleCode/).-->
    <sch:rule context="//sml:PhysicalComponent/sml:contacts/sml:ContactList/
                       sml:contact/gmd:CI_ResponsibleParty/gmd:role/
                       gmd:CI_RoleCode">
       <sch:assert test="@codeList=http://www.isotc211.org/2005/resources/
                         Codelist/gmxCodelists.xml#CI_RoleCode/">
          There shall at least be one contact element that provides contact
          information about the principal investigator.
       </sch:assert>
       <sch:assert test="@codeListValue=CI_RoleCode_principalInvestigator">
          There shall at least be one contact element that provides contact
          information about the principal investigator.
       </sch:assert>
    </sch:rule>
    <!--The SensorML document shall describe the observed property/
        properties.-->.
      <sch:rule context="//sml:PhysicalSystem">
       <sch:assert test="count(sml:outputs/sml:OutputList/sml:output) >= 1">
          Error: At least one observed property shall be included as output in
          the description of the sensor.
       </sch:assert>
    </sch:rule>
    <!--For each output of a sensor, there shall be a definition attribute
        describing the meaning of the output.-->
    <sch:rule context="//sml:sml:PhysicalSystem/sml:outputs/sml:OutputList/
                       sml:output/*">
       <sch:assert test="@definition">
          For each output of a sensor, there shall be a definition attribute
          describing the meaning of the output.
       </sch:assert>
    </sch:rule>
    <!--The location of the sensor must be described.-->
    <sch:rule context="//sml:PhysicalSystem">
       <sch:assert test="sml:position">
          Error: The location of the sensor is missing
       </sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```

Rules for Command Description

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
            xmlns:sml="http://www.opengis.net/sensorml/2.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:swe="http://www.opengis.net/swe/2.0"
            xmlns:gml="http://www.opengis.net/gml"
            xmlns:xlink="http://www.w3.org/1999/xlink"
            xsi:schemaLocation="http://www.opengis.net/sensorml/2.0
                                http://schemas.opengis.net/sensorML/2.0/
                                sensorML.xsd"
            schemaVersion="ISO19757-3">
   <ns prefix="sml" uri="http://www.opengis.net/sensorml/2.0"/>
   <ns prefix="gml" uri="http://www.opengis.net/gml"/>
   <ns prefix="swe" uri="http://www.opengis.net/swe/2.0"/>
   <ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
   <ns prefix="sch" uri="http://purl.oclc.org/dsdl/schematron"/>
   <sch:pattern id="SystemValidation">
      <!--There shall be at least one unique identifier for referring to the
          command.-->
      <sch:rule context="//sml:SimpleProcess">
         <sch:assert test="count(sml:identification/sml:IdentifierList/
                           sml:identifier) >= 1">
            There shall be at least one unique identifier for referring to the
            command.
         </sch:assert>
      </sch:rule>
      <!--The term element within an identifier of a command shall have a
          definition attribute which refers to a vocabulary entry defining the
          meaning of the identifier.-->
      <sch:rule context="//sml:SimpleProcess/sml:identification/
                         sml:IdentifierList/sml:identifier/sml:Term">
         <sch:assert test="@definition">
            The term element within an identifier of a command shall have a
            definition attribute which refers to a vocabulary entry defining the
            meaning of the identifier.
         </sch:assert>
      </sch:rule>
      <!--There shall be at least one classifier describing the nature of the
          command (e.g. data access, configuration change, etc.).-->
      <sch:rule context="//sml:SimpleProcess">
         <sch:assert test="count(sml:classification/sml:ClassifierList/
                           sml:classifier) >= 1">
            There shall be at least one unique identifier for referring to the
            command.
         </sch:assert>
      </sch:rule>
      <!--The term defining the classifier of a command shall have a definition
          attribute which refers to a vocabulary entry defining the meaning of the
          classifier..-->
      <sch:rule context="//sml:SimpleProcess/sml:classification/
                         sml:ClassifierList/sml:classifier/sml:Term">
         <sch:assert test="@definition">
            The term defining the classifier shall have a definition attribute
            which refers to a vocabulary entry defining the meaning of the
            classifier.
         </sch:assert>
      </sch:rule>
      <!--For each input there shall be a definition describing its meaning..-->
      <sch:rule context="//sml:SimpleProcess/sml:inputs/sml:InputList/
                         sml:input/sml:DataInterface/sml:data/swe:DataStream/
                         swe:elementType/swe:DataRecord/swe:field/*">
```

```xml
        <sch:assert test="@definition">
           For each input there shall be a definition describing its meaning.
        </sch:assert>
     </sch:rule>
     <!--Within the inputs section a SensorML 2.0 compliant encoding element shall
         be provided that defines how the input parameters shall be encoded when
         they are transmitted to the sensor.-->
     <sch:rule context="//sml:SimpleProcess">
        <sch:assert test="count(/sml:inputs/sml:InputList/sml:input/
                          sml:DataInterface/sml:data/swe:DataStream/
                          swe:encoding) >= 1">
           Within the inputs section a SensorML 2.0 compliant encoding element
           shall be provided that defines how the input parameters shall be
           encoded when they are transmitted to the sensor.
        </sch:assert>
     </sch:rule>
  </sch:pattern>
</sch:schema>
```

Rules for Output Description

```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
            xmlns:sml="http://www.opengis.net/sensorml/2.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:swe="http://www.opengis.net/swe/2.0"
            xmlns:gml="http://www.opengis.net/gml"
            xmlns:xlink="http://www.w3.org/1999/xlink"
            xsi:schemaLocation="http://www.opengis.net/sensorml/2.0
                                http://schemas.opengis.net/sensorML/2.0/
                                sensorML.xsd"
            schemaVersion="ISO19757-3">
   <ns prefix="sml" uri="http://www.opengis.net/sensorml/2.0"/>
   <ns prefix="gml" uri="http://www.opengis.net/gml"/>
   <ns prefix="swe" uri="http://www.opengis.net/swe/2.0"/>
   <ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
   <ns prefix="sch" uri="http://purl.oclc.org/dsdl/schematron"/>
   <sch:pattern id="SystemValidation">
      <!--For each output of a command there shall be a definition describing its
          meaning.-->
      <sch:rule context="//sml:SimpleProcess/sml:outputs/sml:OutputList/
                         sml:output/sml:DataInterface/sml:data/swe:DataStream/
                         swe:elementType/swe:DataRecord/swe:field/*">
         <sch:assert test="@definition">
            For each output there shall be a definition describing its meaning.
         </sch:assert>
      </sch:rule>
      <!--Within the outputs section a SensorML 2.0 compliant encoding element
          shall be provided that defines how the output data is encoded when
          returned by the sensor.-->
      <sch:rule context="//sml:SimpleProcess">
         <sch:assert test="count(/sml:outputs/sml:OutputList/sml:output/
                           sml:DataInterface/sml:data/swe:DataStream/
                           swe:encoding) >= 1">
            Within the inputs section a SensorML 2.0 compliant encoding element
            shall be provided that defines how the input parameters shall be
            encoded when they are transmitted to the sensor.</sch:assert>
      </sch:rule>
   </sch:pattern>
</sch:schema>
```