D2.9 Draft Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development

<table>
<thead>
<tr>
<th>Title</th>
<th>D2.9 Draft Guidelines for the use of Observations &amp; Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creator</td>
<td>INSPIRE Cross Thematic Working Group on Observations &amp; Measurements</td>
</tr>
<tr>
<td>Date</td>
<td>2013-02-22</td>
</tr>
<tr>
<td>Subject</td>
<td>Use of Observations &amp; Measurements and Sensor Web Enablement-related standards in INSPIRE</td>
</tr>
<tr>
<td>Publisher</td>
<td>INSPIRE Cross Thematic Working Group on Observations &amp; Measurements</td>
</tr>
<tr>
<td>Type</td>
<td>Text</td>
</tr>
<tr>
<td>Description</td>
<td>This document describes the use of Observations &amp; Measurements and Sensor Web Enablement-related standards in INSPIRE data specification development. This version (version 2, release candidate 3) reflects the content of the draft amendment to Commission Regulation (EU) No 1089/2010 for the Annex II+III spatial data themes as submitted to the INSPIRE Committee.</td>
</tr>
<tr>
<td>Contributor</td>
<td>Members of the INSPIRE Cross Thematic Working Group on Observations &amp; Measurements</td>
</tr>
<tr>
<td>Format</td>
<td>Portable Document Format (pdf)</td>
</tr>
<tr>
<td>Source</td>
<td>Public</td>
</tr>
<tr>
<td>Rights</td>
<td>Public</td>
</tr>
<tr>
<td>Identifier</td>
<td>D2.9_v2.0rc3</td>
</tr>
<tr>
<td>Language</td>
<td>En</td>
</tr>
<tr>
<td>Coverage</td>
<td>Project duration</td>
</tr>
</tbody>
</table>
O&M Guidelines – Executive Summary

While INSPIRE is foremost a Spatial Data Infrastructure, several of the Annex Themes have been specified so that their scope, in addition to the basic spatial information, includes measured, modelled or simulated data. The ISO 19156:2011 standard on Observations and Measurements (O&M) was designed for this explicit purpose, and thus shall be used in INSPIRE to cover these requirements.

The following INSPIRE themes have identified O&M as integrally relevant to their thematic domain and are including elements of O&M into their data specifications:

- Geology
- Oceanographic geographical features
- Atmospheric conditions and Meteorological geographical features
- Environmental monitoring facilities
- Soil

In addition to these themes, several further INSPIRE themes have been identified to which observational information, while not at the core of the data specification, is relevant. These themes are:

- Area management/restriction/regulation zones and reporting units: Not mentioned but relevant for reporting on aggregated levels
- Human Health and Safety: provision of health determinants
- Land cover: Observations form the basis for land cover information
- Natural risk zones: Not in model, but use case states: "Monitoring data:
  - Type of monitoring instrumentation
  - Location of sampling measurements
  - Type and record of measurements"
- Production and industrial facilities: Relevant for provision of emissions data for E-PRTR
- Statistical units & Population distribution, demography: StatisticalDataValue looks a bit like OM_Observation, SU would also make sense
- Utility and governmental services: Currently stated that "Non-geographic data (e.g. information on flow in m³/s) is also out of scope of this specification"
- Habitats and biotopes & Species distribution: Observation form the basis for these themes, link to primary observations

While the O&M standard provides a generic framework for the provision of measurement data, there are many ways of utilizing the core structures. In order to assure compatibility across thematic tailoring versions of the O&M standards, the X-TWG OM has provided guidelines as to how this standard is to be used within INSPIRE. These guidelines should be taken into account in all INSPIRE themes integrating or referencing to the O&M standard.
Acknowledgements

Many individuals and organisations have contributed to the development of these Guidelines.

The Cross-Thematic Working Group on Observations and Measurements (X-TWG-OM) included: Katharina Schleidt (Coordinator), Jandirk Bulens, Simon Cox, Sylvain Grellet, Huibert-Jan Lekkerkerk, Dominic Lowe, Michael Lutz, Clemens Portele, Ilkka Rinne, Heino Rudolf, Laszlo Sores, Jeremy Tandy, Spiros Ventouras, Gavin Walker, Bruce Wright, Alessandro Sarretta (European Commission contact point till May 2012), Tomáš Rezník (European Commission contact point from May till August 2012), Vlado Cetl (European Commission contact point from August 2012)

Contact information
Vanda Nunes de Lima
European Commission Joint Research Centre
Institute for Environment and Sustainability
Spatial Data Infrastructures Unit
TP262, Via Fermi 2749
I-21027 Ispra (VA)
ITALY
E-mail: vanda.lima@jrc.ec.europa.eu
Tel.: +39-0332-7865052
Fax: +39-0332-7866325
http://ies.jrc.ec.europa.eu/
http://ec.europa.eu/dgs/jrc/
http://inspire.jrc.ec.europa.eu/
# Table of contents

1. Scope ........................................................................................................... 1

2. Normative references .................................................................................. 1

3. Terms and abbreviations ............................................................................ 1
   3.1 Terms ....................................................................................................... 1
   3.2 Abbreviations .......................................................................................... 4

4. A Short Introduction to O&M and SWE ....................................................... 4
   4.1 Short introduction to O&M ...................................................................... 4
   4.1.1 Observations ....................................................................................... 4
   4.1.2 Sampling ............................................................................................. 8
   4.2 Short introduction to Sensor Web Enablement (SWE) ........................... 10
   4.2.1 SOS Overview .................................................................................. 10
   4.2.2 SensorML Overview .......................................................................... 12
   4.2.3 SWE Common Result Encoding ....................................................... 12

5. INSPIRE O&M Design Patterns ................................................................. 12
   5.1Use of O&M vs. Coverage Model ............................................................ 12
   5.1.1 Result/Coverage-centric view ............................................................ 13
   5.1.2 Observation-centric view ................................................................... 13
   5.2 Observation-centric view Design Patterns by Feature of Interest
       Dimensionality ....................................................................................... 14
   5.2.1 Feature of Interest Point ................................................................... 14
   5.2.2 Feature of Interest Curve .................................................................. 16
   5.2.3 Feature of Interest Surface ............................................................... 17
   5.3 Observation-centric view – Specimen ..................................................... 19
   5.3.1 Single Result in Time ....................................................................... 19
   5.3.2 Multiple Results in Time ................................................................... 19
   5.4 Coverage vs. Non-Coverage Observation Results ................................. 20
   5.4.1 Coverage results ................................................................................ 20
   5.4.2 Non-Coverage Results ....................................................................... 22
   5.5 Decision Tree ......................................................................................... 23

6. O&M Extensions to INSPIRE ..................................................................... 24
   6.1 General.................................................................................................... 24
   6.2 Feature of Interest .................................................................................. 25
   6.2.1 SamplingFeature vs. Domain Feature ............................................... 26
   6.2.2 Sampling Features in SamplingCoverageObservations ........................ 26
   6.2.3 SamplingFeature vs. SampledFeature ............................................... 27
   6.3 Observed Property .................................................................................. 28
   6.3.1 AbstractObservableProperty ............................................................. 29
   6.3.2 ObservableProperty .......................................................................... 29
   6.3.3 Statistical Measures ......................................................................... 30
   6.3.4 Constraints ....................................................................................... 30
   6.3.5 CompositeObservableProperty ....................................................... 30
   6.3.6 Feature catalogue ............................................................................. 30
   6.4 Process .................................................................................................... 38
   6.4.1 Process Parameters .......................................................................... 39
   6.4.2 Process Encoding with SensorML ..................................................... 41
   6.4.3 Feature catalogue ............................................................................. 42
   6.5 Result ..................................................................................................... 45
   6.5.1 State of the Art .................................................................................. 45
   6.5.2 SWE Result Encoding ....................................................................... 46
   6.5.3 GML Coverage Result Encoding ..................................................... 48
   6.5.4 Out-of-Band Result Encoding ........................................................... 48
Figures

Figure 1: The basic Observation type ................................................................. 5
Figure 2: Sampling feature vs. sampled feature .................................................. 8
Figure 3: Basic Sampling Types .......................................................................... 9
Figure 4: Observations Package ....................................................................... 24
Figure 5: High Level SampledFeature .............................................................. 27
Figure 6: Complex Properties Model .............................................................. 29
Figure 7: Process Class ..................................................................................... 39
Figure 8: Results - Simple Components ............................................................ 46
Figure 9: Results - Range Components ............................................................. 47
Figure 10: Specialised Observation Types ....................................................... 50
Figure 11: PointObservation ........................................................................... 51
Figure 12: PointTimeSeriesObservation .......................................................... 53
Figure 13: MultiPointObservation ................................................................... 54
Figure 14: Profile Observation ......................................................................... 56
Figure 15: TrajectoryObservation ................................................................... 57
Figure 16: GridObservation ............................................................................. 59
Figure 17: GridSeriesObservation ................................................................... 60
Figure 18: Direct association to observations and observation sets ..................... 67
Figure 19: Linkage with parameter attribute .................................................... 68
Figure 20: Observing Capability Class ............................................................. 70
1 Scope

This document provides guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development.

These guidelines provide the basis for the definition of INSPIRE Annex II and III data specifications if they are using the O&M schema.

2 Normative references

ISO 19115:2003, Geographic information — Metadata
ISO 19115:2003/Cor 1:2006, Geographic information — Metadata — Corrigendum 1
ISO 19123:2005, Geographic information — Schema for coverage geometry and functions
ISO 19136:2007, Geographic information — Geography Markup Language
ISO 19156:2011, Geographic information — Observations and measurements

3 Terms and abbreviations

3.1 Terms

For the purposes of this document, the following terms and definitions apply.

(1) application schema
conceptual schema for data required by one or more applications

[ISO 19101:2002, definition 4.2]

(2) coverage
feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain

EXAMPLE Examples include a raster image, polygon overlay or digital elevation matrix.

NOTE In other words, a coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

[ISO 19123:2005, definition 4.1.7]

(3) data type
specification of a value domain with operations allowed on values in this domain

[ISO/TS 19103:2005, definition 4.1.5]

EXAMPLE Integer, Real, Boolean, String, Date (conversion of a date into a series of codes).
NOTE Data types include primitive predefined types and user-definable types. All instances of a data type lack identity.

[ISO 19156:2011]

(4) **domain feature**
feature of a type defined within a particular application domain

NOTE This may be contrasted with observations and sampling features, which are features of types defined for cross-domain purposes.

[ISO 19156:2011(E)]

(5) **ex-situ**
referring to the study, maintenance or conservation of a specimen or population away from its natural surroundings

NOTE Opposite of in-situ.

[ISO 19156:2011(E)]

(6) **feature**
abstraction of real-world phenomena


NOTE A feature may occur as a type or an instance. In this document, feature instance is meant unless otherwise specified.

[ISO 19156:2011(E)]

(7) **feature type**
class of features having common characteristics

[ISO 19156:2011(E)]

(8) **measure**
value described using a numeric amount with a scale or using a scalar reference system

[ISO 19136:2007, definition 4.1.41]

(9) **measurement**
set of operations having the object of determining the value of a quantity


(10) **observation**
act of measuring or otherwise determining the value of a property

[ISO 19156:2011(E)]

(11) **observation procedure**
method, algorithm or instrument, or system of these, which may be used in making an observation

[ISO 19156:2011(E)]

(12) **observation protocol**
combination of a sampling strategy and an observation procedure used in making an observation
(13) **observation result**
estimate of the **value** of a **property** determined through a known observation procedure

(14) **property**
facet or attribute of an object referenced by a name

EXAMPLE Abby's car has the colour red, where "colour red" is a property of the car instance

(15) **property type**
characteristic of a **feature type**

EXAMPLE cars (a feature type) all have a characteristic colour, where "colour" is a property type

NOTE 1 The value for an instance of an observable property type can be estimated through an act of observation

NOTE 2 In chemistry-related applications, the term "determinand" or "analyte" is often used.

(16) **sampling feature**
**feature** which is involved in making **observations** concerning a **domain feature**, this feature provides the direct context for the specific observation (spatial, specimen)

EXAMPLE station, transect, section or specimen.

NOTE A sampling feature is an artefact of the observational strategy, and has no significance independent of the observational campaign.

(17) **spatial sampling feature**
a sampling feature with a spatial coverage. Used for observations where the result varies within the scope of the feature

EXAMPLE ShipsTrack, Profile, Swath.

NOTE A spatial sampling feature is an artefact of the observational strategy, and has no significance independent of the observational campaign.

(18) **specimen**
a sampling feature that is a physical sample, obtained for ex-situ observations, usually analysed in a laboratory. A specimen may be archived for future reference after the observation has been performed.

(19) **value**
element of a type domain

NOTE 1 A value considers a possible state of an object within a class or type (domain).
NOTE 2 A data value is an instance of a datatype, a value without identity.

NOTE 3 A value can use one of a variety of scales including nominal, ordinal, ratio and interval, spatial and temporal. Primitive datatypes can be combined to form aggregate datatypes with aggregate values, including vectors, tensors and images.

[ISO 19156:2011(E)]

### 3.2 Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FoI</td>
<td>Feature of Interest</td>
</tr>
<tr>
<td>GFM</td>
<td>General Feature Model</td>
</tr>
<tr>
<td>GML</td>
<td>Geography Markup Language</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>Observations and Measurements</td>
</tr>
<tr>
<td>OGC</td>
<td>Open Geospatial Consortium</td>
</tr>
<tr>
<td>SensorML</td>
<td>Sensor Model Language</td>
</tr>
<tr>
<td>SOS</td>
<td>Sensor Observation Service</td>
</tr>
<tr>
<td>SWE</td>
<td>Sensor Web Enablement</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>

### 4 A Short Introduction to O&M and SWE

#### 4.1 Short introduction to O&M

While INSPIRE is foremost a Spatial Data Infrastructure, several of the Annex Themes have been specified so that their scope, in addition to the basic spatial information, includes measured, modelled or simulated data about the real world. The ISO 19156:2011 standard on Observations and Measurements (O&M) was designed for this explicit purpose, and thus shall be used in INSPIRE to cover these requirements.

This section serves as a simple overview of the main concepts of the O&M standard for better understanding of the INSPIRE specific design patterns proposed. For more detailed information on O&M, please refer either to ISO 19156:2011 or the OGC Document: OGC 10-004r3 (Geographic Information: Observations and Measurements - OGC Abstract Specification Topic 20)

#### 4.1.1 Observations

An Observation is an action whose **result** is an estimate of the value of some **property** of the **feature-of-interest**, at a specific point in **time**, obtained using a specified **procedure**

after Cox 2008

For the structuring of data pertaining to observations, the O&M standard has defined the OM_Observation type. The following diagram shows the basic OM_Observation type with all its attributes, associations and constraints.
4.1.1.1. Observation Associations

The following associations link classes to the observation that serve to explicitly describe all aspects of the observation performed,

4.1.1.1.1. Feature-Of-Interest

Association: Domain
Role: featureOfInterest
Inverse Role: propertyValueProvider

The association Domain links to the GFI_Feature that is the subject of the observation. This is a representation of the real world object the property is being estimated on.

In the design patterns presented in chapter "5 INSPIRE O&M Design Patterns", the following types are used as Feature-Of-Interest:
- species occurrence point
- air bubble surrounding intake
- water column
- trajectory
- grid
- water sample

The following terms are used to refer to the Feature-Of-Interest in other domains:

Earth Observations: 2-D swath or scene; 3-D sampling space
Earth science simulations: Section, swath, volume, grid
Assay/Chemistry: Sample
Geology field observations: Location of structure observation; Rock sample

4.1.1.1.2. Property

Association: Phenomenon
Role: observedProperty

The association Phenomenon links to the GF_PropertyType describing the property of the feature-of-interest that is being estimated in this observation.

In the design patterns presented in chapter “5 INSPIRE O&M Design Patterns”, the following types are used as observedProperty:

- species identification
- height
- O₃ hourly mean
- chlorophyll content
- salinity
- (water) temperature
- (water) color
- nitrate concentration
- biochemical oxygen demand

The following terms are used to refer to the property in other domains:

Earth Observations: parameter, variable
Metrology: measurand
Earth science simulations: Variable, parameter
Assay/Chemistry: Analyte
Geology field observations: Strike and dip, lithology, alteration state, etc.

4.1.1.1.3. Procedure

Association: ProcessUsed
Role: procedure
Inverse Role: generatedObservation

The association ProcessUsed links to the OM_Process describing the process or methodology used in the estimation of the result in this observation.

The following terms are used to refer to the procedure in other domains:

Earth Observations: method, sensor
Metrology: instrument
Earth science simulations: Earth process simulator
Assay/Chemistry: Instrument, analytical process

4.1.1.1.4. Result

Association: Range
Role: result

The association Range links to the estimate of the property on the feature-of-interest generated by the procedure.

The following terms are used to refer to the Result in other domains:
Earth Observations: observation value, measurement value, observation
Metrology: value
Earth science simulations: model
Assay/Chemistry: Analysis

4.1.1.2. Observation Attributes

The following attributes provide further detail on the observation performed.

4.1.1.2.1. Parameter

Datatype: NamedValue

The attributes provided under parameter describe any event-specific parameter of relevance to interpreting the observation. These will typically complement the re-usable (event-neutral) description of the observation procedure. The datatype NamedValue allows for the provision of key/value pairs for the parameter.

Within O&M, any GenericName may be provided as the name, and any data type may be provided for the value. In INSPIRE, this has been further constrained; for more information, see section “6.4.1 Process Parameters”.

4.1.1.2.2. PhenomenonTime

Datatype: TM_Object

The attribute phenomenonTime shall describe the time that the result applies to the property of the feature-of-interest. This may be the time when a specimen was collected or the observation procedure was performed on a real-world feature, but may be in the future in the case of forecasts, or in the deep past for archeological or geological observations. The type ‘TM_Object’ allows for time instants, time periods (where the result is extensive in time such as a temporal coverage), or temporal topologies if this is the most appropriate representation.

4.1.1.2.3. ResultQuality

Datatype: DQ_Element

When the Observation result consists of a single value, or a set of values that are all of the same quality, the quality of the result is to be provided here. However, in the case of complex results (spatial or temporal coverages), the quality may vary across the result values; in this case, the quality should be provided for each value within the result.

4.1.1.2.4. ResultTime

Datatype: TM_Instant

The attribute resultTime describes the time when the result became available, typically when the procedure associated with the observation was completed. For some observations this is identical to the phenomenonTime. However, there are important cases where they differ; an example of this is a specimen analyzed in a laboratory, where the PhenomenonTime should correspond to the time the specimen was taken, while the ResultTime is the time when the laboratory analysis was completed.

4.1.1.2.5. ValidTime

Datatype: TM_Period

If present, the attribute validTime describes the time period during which the result is intended to be used. This attribute is commonly required in forecasting applications.
4.1.2 Sampling

In order to understand the meaning of an observation, one must also understand the exact domain of the observation, the feature-of-interest. In some cases, this feature-of-interest can be analysed in its entirety. Some examples for this are:

- The weight of a specific animal
- The type of crop planted on a specific field

In this case, this domain feature should be directly used as the feature-of-interest of the Observation. In other cases, it is difficult to estimate a property on the entirety of a feature; thus, one usually samples one representative part of this larger feature for analysis purposes. In this case, the feature-of-interest is a sample which in O&M is called the samplingFeature, which refers to the feature it has been taken as a representative of as its sampledFeature. Some examples of this are:

- A rock sample taken as a representative for an outcrop.
- The measurement of air temperature at a particular location (sampling the atmosphere at a point).
- A sample of river water taken as a representative for the river segment sampled from.

The diagram below shows the relation between SamplingFeature and SampledFeature:

![Diagram](image)

**Figure 2: Sampling feature vs. sampled feature**

In further cases, a specific specimen is taken and analysed; for this purpose, SF_Specimen has been defined as a specialised form of SF_SamplingFeature.
4.1.2.1. SamplingFeature

A samplingFeature is an intermediate feature involved in an observation which allows an estimate of a property value for the ultimate feature of interest to be made; this feature provides the direct context for the specific observation (spatial, specimen). In the case of a spatial sampling feature, which is derived from the SF_SamplingFeature, the spatialSamplingFeature is a point, line, surface or volume which may be co-located with the ultimate FOI. e.g. a point in a river. The result values will vary across the spatialSamplingFeature.

The following terms describe specific samplingFeatures in various domains:

Earth Observations: 2-D swath or scene; 3-D sampling space
Earth science simulations: Section, swath, volume, grid
Assay/Chemistry: Polished section, probe spot, pulp
Geology field observations: Outcrop; Location of structure observation

4.1.2.2. Sampled Feature

The sampledFeature is the feature the samplingFeature was sampled from, providing the ultimate context for the observation. An example of sampledFeature would be the river segment a specimen was taken from.

The following terms are used to refer to the sampledFeature in other domains:
Earth Observations: Earth surface; media (air, water, ...), Global Change
Earth Science simulations: Master Directory "Topic"
Geology field observations: Atmosphere, ocean, solid earth

4.1.2.3. Specimen

A specimen is a feature sampled from a feature of interest to enable ex-situ observation, such as in a laboratory. Information on how the specimen was acquired, where it is presently stored, and its preparation procedure are provided.

The following terms are used to refer to the Specimen in other domains:

- Assay/Chemistry: Sample; Pulp, separation
- Geology field observations: Rock sample

4.2 Short introduction to Sensor Web Enablement (SWE)

In addition to the use of the Observations and Measurements standard, further elements of the OGC Sensor Web Enablement Suite (SWE) have been identified as useful for the encoding and provision of observational data. While further SWE specifications may be nominated for use in INSPIRE, at the present we have identified the following:

- Sensor Observation Service (SOS): service created for the provision of observational data;
- SensorML: Standard for the provision of procedural information;
- SWE Common: Includes result encoding options.

Please note, that there are more SWE elements and services respectively. However, they are not within the scope of this document. Please see [http://www.opengeospatial.org/ogc/markets-technologies/swe](http://www.opengeospatial.org/ogc/markets-technologies/swe) for a wider view on this topic including sensor tasking, filtering, notifications from sensor measurements, etc.

4.2.1 SOS Overview

The OGC SOS specification is based on the OGC Web Service Common specification, thus, it has shares structures and data types of service requests with the other OGC Web Services. In the case of SOS, the operation signature is constrained by the observation schema, as it defines the response model. The following SOS operations have been identified as relevant to INSPIRE, and will be simply described in the sections below:

- GetCapabilities
- GetObservation
- DescribeSensor
- GetFeatureOfInterest

As the SOS 2.0 standard has been approved as an official OGC standard, we strongly advise users providing INSPIRE data via a SOS to use this new and updated version. For more detailed information on SOS version 2.0, please refer to the OGC Document [http://www.opengis.net/doc/IS/SOS/2.0 (OGC® Sensor Observation Service Interface Standard)](http://www.opengis.net/doc/IS/SOS/2.0) available at [https://portal.opengeospatial.org/files/?artifact_id=47599](https://portal.opengeospatial.org/files/?artifact_id=47599).

4.2.1.1. GetCapabilities

This operation allows clients to retrieve service metadata about a specific service instance, including metadata about the tightly-coupled data served. In addition to more generic capabilities response
elements such as filter options, the SOS GetCapabilities returns a list of so called Offerings, which are groupings of available observations described by their feature of interest, procedure, observed property, temporal coverage and the like. This allows the user application to clearly identify the types and quality of data that can be requested from this service.

Please note, that SOS in version 1.0 does not restrict creation of observation offerings. On the other hand, SOS 2.0 is more specific and defines that each observation offering is limited to be associated with exactly one sensor (system). This solves the issue of the SOS 1.0 with ambiguous groupings of observations to offerings.

4.2.1.2. GetObservation

The GetObservation operation is designed to query a service to retrieve observation data structured according to the Observations and Measurements specification. Within the GetObservation request, the user can provide a filter, including the desired observed property, feature of interest and time interval, which determines which observations are to be returned. The response to the GetObservation request is one or more observations encoded as an OM_Observation, as shown in section “4.1.1 Observations”.

4.2.1.3. DescribeSensor

DescribeSensor is designed to request detailed sensor metadata. The response of this operation provides this procedural metadata, encoded using a well-defined format. The SOS 2.0 thereby recommends the usage of the Sensor Model Language (SensorML 1.0.1) to encode sensor metadata. This provides all information required on how the observation or measurement was obtained. This information should be sufficient to determine if the data from the observations meets ones further processing requirements.

4.2.1.4. GetFeatureOfInterest

GetFeatureOfInterest returns a feature of interest that is associated with an observation served by the SOS. The features are encoded using a specific GML application schema.

4.2.1.5. Comparison of SOS 1.0 and 2.0

SOS V2.0 has been ratified as an OGC standard in March 2012. This section briefly describes the differences between SOS 1.0 and SOS 2.0 to ease the implementation of INSPIRE themes dealing with the O&M concept. The major changes may be summed up as followings:

- The specification is based upon GML 3.2.1, Filter Encoding 2.0, SWE Common 2.0, O&M 2.0 and the SWE Service Model. This makes SOS 2.0 closer to the INSPIRE recommended encoding.
- The behaviour of all operations was revised and defined in more detail.
- The standard defines minimum capabilities a SOS needs to support with respect to spatial and temporal filter operations and operands.
- A spatial filtering profile considerably improves spatial filtering of observation data.
- A KVP binding for core operations is defined.
- The standard defines a SOAP binding.
- Offerings may no longer be associated with multiple sensors – only one sensor per offering is allowed which simplifies and clarifies observation management for offerings.
• The application of a property inheritance mechanism reduces the amount of redundant information in a GetCapabilities document.

• The GetCapabilities document no longer lists all features of interest belonging to the observations stored by the service, which could previously lead to an explosion of the document's size. The GetFeatureOfInterest operation now serves as the primary way to retrieve information on features of interest.

• Efficient insertion of result data was added as a new capability – previously only result retrieval was considered.

As SOS V2.0 has many good extensions and modifications, it would be advantageous to implement INSPIRE services using this new version. Within the OGC Web Service Testbed 9 (http://www.opengeospatial.org/projects/initiatives/ows-9) an automatic test suite as well as a SOS 2.0 reference implementation are being implemented. They will be available in their final version by December 2012.

4.2.2 SensorML Overview

Sensor Model Language (SensorML) was created by the sensor community for structuring of information pertaining to sensors. This covers all procedural information as required within O&M, and is often the default procedure format expected by SOS implementations as this is the primary recommendation from SOS 2.0. Of primary interest to these guidelines is the SensorML component class – this class was created for the description of measurement components (i.e. individual sensors). For complex properties the system class may be used.

While SensorML was primarily designed by the sensor community, due to its abstract structure it can also be used for data survey process descriptions not using sensors, or very abstract concepts of sensors such as human sensors performing a biodiversity survey.

At present, SensorML is only available in the version 1.0.1. Work on the 2.0 version is ongoing, but it is at present not clear how this will differ from the 1.0 version or when it will be made available. For more detailed information on SensorML, please refer to the OGC Document: OGC® 07-000 (OpenGIS® Sensor Model Language (SensorML) Implementation Specification)

A mapping between the INSPIRE Process and SensorML is provided in section “6.4.2 Process Encoding with SensorML”

4.2.3 SWE Common Result Encoding

OGC SWE 2.0 provides dedicated result types in the package SWE Common. For an example of the encoding of time series data (in the example given it pertains to air quality measurements), please see Annex “A.1 SWE Common Results”.

5 INSPIRE O&M Design Patterns

5.1 Use of O&M vs. Coverage Model

Many types of spatial data can be structured using either O&M or GML coverages. As an initial step one must determine which model to follow for specifying the data models. In certain cases, often pertaining to coverage results, the result is of primary interest while the methodology used in attaining this result is secondary. In other cases, while the result is still of importance, a good understanding of the process that was utilized in generating these results is of utmost importance in proper further
utilization of the result data. Differentiation in a Result/Coverage-centric vs. an Observation-centric view helps us to determine if a specific type of data should be encoded via O&M or GML coverages.

5.1.1 Result/Coverage-centric view

In the Result/Coverage-centric view, the result (generally a coverage in this case) is the primary object of interest while the description of the observation process is just metadata of the result; coverages are “first class citizens” and information on the observation is metadata about the coverage.

In this context, it is possible to envision design patterns that forgo provision of procedural metadata entirely as this is not of further relevance for the interpretation of the result. However, there are also cases where the procedural metadata can be of relevance to the further processing chain. In such cases, the coverage metadata can be used to supply such information.

If a Result/Coverage-centric view is best suited for modelling a specific domain, then O&M is not relevant for this purpose. Please see the relevant ISO 19123:2005 Geographic information -- Schema for coverage geometry and functions.

5.1.2 Observation-centric view

In the Observation-centric view, observations are first-class citizens, and the result could almost be seen as “metadata”; full knowledge of the result acquisition process is necessary and must be provided. In this context, design patterns fully embracing the richness of O&M are recommended, as these are far richer in properties and description of the observation process.

If an Observation-centric view is best suited for modelling a specific domain, then O&M is relevant for this purpose. Please see the following sections for further information on possible design patterns suited for this purpose.
5.2 Observation-centric view Design Patterns by Feature of Interest Dimensionality

In order to provide a better overview of the various O&M Design Patterns used in INSPIRE, we have structured these by the dimensionality of the featureOfInterest of the observation. A further point of differentiation lies in the result; when the result is provided for a single point in time the dimensionality of the result is usually equivalent to that of the featureOfInterest, when the result is a time series the result has this additional dimension.

5.2.1 Feature of Interest Point

5.2.1.1 Single Result in Time

The simplest O&M Design Pattern provides a single result on a single point. This result value can be a measurement on the featureOfInterest, but can also be a categorisation or other type of observation on the featureOfInterest. The most common examples of this design pattern pertain to biodiversity observations pertaining to individuals.

As a first example, we will demonstrate how an individual of a specific species can be identified; in this example an individual tree of species *Abies alba*. In this example, the featureOfInterest will be defined as a “species occurrence point”. In order to understand this example it is important to understand that we cannot model our featureOfInterest directly as an occurrence of *Abies alba*, as this identification is actually an observation on the individual; at a later time it may be determined that this individual is actually of a different species, which can then be provided as an updated observation. In this example, the observedProperty is “Species Identification”, observed using the procedure “Expert Judgement”. The result then refers to the correct entry in a standardised species list, in this case to “*Abies alba*”.

As a second example, we demonstrate how a measurement performed on an individual can be structured; in this example, the height of an individual tree. The featureOfInterest will again be defined as a “species occurrence point”; it could be the same “species occurrence point” instance for which we provided an identification in the previous example. In this example, the observedProperty is “Height”, observed using the procedure “Magic Triangulation”. The result then provides the height of the tree in meters.
5.2.1.2. Multiple Results in Time

In other cases, repeated measurements are taken on a single point in space; this set of measurement is referred to as a time series. A time series may be automatically generated by a sensor taking regular measurements, but could also stem from repeated manual measurements.

As a first example, we show how multiple measurements over time can be structured; in this case, we show an air quality monitoring station providing hourly ozone measurements. The featureOfInterest represents the direct surrounds of the air intake (i.e. the air bubble surrounding the air intake, to be modelled as a sampling point (SF_SamplingPoint) and is modelled as a samplingFeature. The location for the measurements is provided through the FoI. As this design pattern usually provides a time series (temporal coverage) result, the phenomenonTime and/or resultTime will often be provided together with the result values. Administrative information relevant to the facility is stored in the EnvironmentalMonitoringFacility Class from the Environmental Monitoring Facilities Application Model.

As a second example, we show how multiple manual measurements over time can be structured. This example is related to the example above on the height of a tree. Whereas in the previous example, there was only one height measurement performed, in this example, the height of the tree is measured every year. The featureOfInterest will again be defined as a “species occurrence point”; it could be the same “species occurrence point” instance for which we provided an identification in the previous example. In this example, the observedProperty is “Height”, observed using the procedure “Magic Triangulation”. The result then provides the height of the tree in meters, with a timestamp (year) provided with each height measurement.
As a final example, we show how multiple observations are taken on an individual that has been selected as a representative (samplingFeature) for a larger group (sampledFeature). In this example, an individual plant is selected from a group of plants. This plant is then tested daily for its chlorophyll content using a non-destructive method. The featureOfInterest will again be defined as a “species occurrence point”. In this example, the observedProperty is “Chlorophyll content”, observed using the procedure “Non destructive method”. The result then provides the chlorophyll content of the individual plant, with a timestamp (day) provided with each chlorophyll measurement.

5.2.2 Feature of Interest Curve

5.2.2.1 Single Result in Time

In this design pattern the featureOfInterest represents a water column at the current location of a moving facility, for example ship measuring the salinity at varying depths along a water column. The water column is modelled as a samplingFeature, with the spatial sampling feature (e.g. a SF_SamplingCurve in O&M) being the media referred to, in this case, the ocean (or a further chain of samplingFeatures finally referring to the media). The actual locations of individual measurements along the water column are provided with the results. All measurement locations should be located within the water column described by the featureOfInterest; depending on the encoding of the result.
the actual location within the `featureOfInterest` as well as `phenomenonTime` and `resultTime` may be provided as a relative (i.e. 5m from start of water column) or absolute (i.e. coordinates including the depth) position.

5.2.2.2. Multiple Results in Time

In this design pattern the `featureOfInterest` represents the trajectory of a moving facility, for example a moving ship making sea surface temperature measurements. The trajectory is modelled as a `samplingFeature`, with the spatial sampling feature (e.g. a `SF_SamplingCurve` in O&M) being the media referred to, in this case, the ocean (or a further chain of `samplingFeatures` finally referring to the media). The actual locations of individual measurements along the trajectory are provided with the results. All measurement locations should be located within the trajectory described by the `featureOfInterest`; depending on the encoding of the result the actual location within the `featureOfInterest` as well as `phenomenonTime` and `resultTime` may be provided as a relative (i.e. 5m from start of trajectory) or absolute (i.e. coordinates) position.

5.2.3 Feature of Interest Surface

5.2.3.1. Single Result in Time

In this design pattern the `featureOfInterest` represents a grid over the ocean; in this example the ocean color will be determined for each grid cell. The boundary of the sampling grid is modelled as a `samplingFeature`, using a `SF_SamplingSurface` that defines the extent of the Grid of data. The actual
locations of individual observations within the grid are provided with the results. All measurement locations should be located within the boundaries described by the featureOfInterest; the grid cell being observed is provided in the domain of the coverage result, the color observed is provided within the range of the coverage result.

5.2.3.2. Multiple Results in Time

In this design pattern the featureOfInterest represents a grid over the ocean; in this example the ocean temperature will be modelled for each grid cell over time. The boundary of the sampling grid is modelled as a samplingFeature, using a SF_SamplingSurface that defines the extent of the Grid of data. The result should be a GML RectifiedGridCoverage or GML ReferenceableGridCoverage containing the grid points (as the spatio-temporal domain of the coverage) and the observed sea surface temperature values (as the rangeSet of the coverage. Note that one of the axes of the grid coverage domain must be a temporal axis as GridSeriesObservation is a type of time series.
5.3 Observation-centric view – Specimen

Mobile stuff is actually non-mobile as the specimen is taken at a discrete point in space and time although the container doing the measurements may be mobile.

5.3.1 Single Result in Time

In this design pattern the featureOfInterest represents a sample or specimen taken from the sampled feature and analysed \textit{ex situ} in an external laboratory. The featureOfInterest is modeled as a \textit{SF_Specimen}; the location pertaining to the measurement is provided by the attribute samplingLocation, the current location of the specimen is provided by the attribute currentLocation. Procedural information is provided through a laboratory report, the location of the laboratory is not relevant. In this design pattern, phenomenonTime and resultTime will both be required, as phenomenonTime provides the time the sample was taken whereas resultTime provides the time the results of the laboratory analysis were made available.

5.3.2 Multiple Results in Time

While in the example above the sample was only analysed once, there are also cases where the sample taken is re-analysed at regular intervals. An example of this is the measurement of the biochemical oxygen demand (BOD) in waste water treatment plants; it is measured by taking one sample and studying BOD evolution over time in a laboratory. While the usual result requested is BOD 5 (5 for 5 days) or BOD 21, in some cases you may require the entire time series.
5.4 Coverage vs. Non-Coverage Observation Results

In domains providing multidimensional results, one basic characteristic determining the O&M design pattern to be used is the nature of the result, as well as the way it is commonly structured within the specific domain. In certain domains it is common to provide multidimensional results in the form of simple series of values; in other domains it is common to use a coverage model for such results.

A coverage is a feature that associates positions within a bounded space (its domain) to feature attribute values (its range). In other words, it is both a feature and a function. Examples include a raster image, a polygon overlay or a digital elevation matrix.

A coverage may represent a single feature or a set of features.

A coverage domain is a set of geometric objects described in terms of direct positions. It may be extended to all of the direct positions within the convex hull of that set of geometric objects. The direct positions are associated with a spatial or temporal coordinate reference system. Commonly used domains include point sets, grids, collections of closed rectangles, and other collections of geometric objects. The geometric objects may exhaustively partition the domain, and thereby form a tessellation such as a grid or a TIN. Point sets and other sets of non-conterminous geometric objects do not form tessellations. Coverage subtypes may be defined in terms of their domains.

Coverage domains differ in both the coordinate dimension of the space in which they exist and in the topological dimension of the geometric objects they contain. Clearly, the geometric objects that make up a domain cannot have a topological dimension greater than the coordinate dimension of the domain. A domain of coordinate dimension 3 may be composed of points, curves, surfaces, or solids, while a domain of coordinate dimension 2 may be composed only of points, curves or surfaces. ISO 19107:2003 defines a number of geometric objects (subtypes of the UML class GM_Object) to be used for the description of features. Many of these geometric objects can be used to define domains for coverages. In addition, ISO 19108:2002 defines TM_GeometricPrimitives that may also be used to define domains of coverages.

The range of a coverage is a set of feature attribute values. It may be either a finite or a transfinite set. Coverages often model many associated functions sharing the same domain. Therefore, the value set is represented as a collection of records with a common schema.

The major difference between the coverage result and the non-coverage result design patterns lies in the selection of the featureOfInterest of the Observation. When using a coverage result, the featureOfInterest tends to be a wider geometry while the grid specifying the location for the individual values is provided in the coverage result. When using a non-coverage result, the featureOfInterest tends to be the specific spatial locations sampled from the wider geometry.

5.4.1 Coverage results

Coverage results are often used in the atmospheric domain. In the example below, the featureOfInterest is a wide swath of land over which modelled CO$_2$ concentration is provided. The actual locations for which the CO$_2$ concentration has been modelled are provided in a coverage result together with concentration values.
In accordance with ISO 19156:2011, when using a SamplingCoverageObservation, a dedicated SamplingFeature must be provided instead of directly referencing the domain feature as featureOfInterest; this SamplingFeature must correspond to the domain of the coverage being provided as a result. This SamplingFeature should then reference the domain feature it represents as its SampledFeature (see section “6.2.2 Sampling Features in SamplingCoverageObservations”).

For example, when considering a SamplingCoverageObservation (e.g. of water quality or surface temperature) of Lake Constance, the SampledFeature would represent a lake named Lake Constance with the geometry bounded by the lake’s shorelines (2D) or by the shorelines and the bathymetry (3D). The SamplingFeature would represent the network of the measurement stations or sensors. The SamplingFeature would either have the geometry of the individual sampling points combined (multipoint), a 2D surface (if the measurements are taken from the lake’s surface or at an equal depth below the surface) or a 3D solid (if the individual sampling depths are of importance).

The coverage-valued result would then have its domainSet within the SamplingFeature’s geometry. If a RectifiedGrid is used, it must be evenly spaced and rectangular in all dimensions. For the geographical CRS this would probably mean some kind of interpolation of the actual sampling points into the result grid. In this case the bounds of the result grid would drive the bounds of the geometry of the SamplingFeature. The diagram below shows the sampling and sampled features for the case of a rectified grid coverage observation in Lake Constance.
NOTE If processing chains are merged and results with different geometries from a sequence of observations and processing steps are linked together the generic OM_Observation class should be used. SamplingCoverageObservation should only be used if correct SamplingFeatures are provided for all observations within the processing chain.

5.4.2 Non-Coverage Results

In the biodiversity domain, individual observations or time series of observations are common. In the example below, the featureOfInterest is a single point selected from a wide swath of land, at which the abundance of grass is being surveyed. Thus, the swath of land, which in the previous example was the featureOfInterest, in this example becomes the sampledFeature, from which the individual survey points have been selected. An observation is created for each individual survey point, and the result, which can be either an individual value or a time series, is provided.
5.5 Decision Tree

The following decision tree may be a support in the determination of the correct model to use for a specific use case:

- **Is the result observation centric?**
  - **no**
  - **yes**
    - **(A) Result/Coverage centric point of view**
      - 1st class citizen: result (often types coverage)
      - 2nd class citizen: observation (coverage metadata)
      - => use coverage and, if needed, coverage metadata
    - **(B) Point of view centered on observation/methodological information**
      - 1st class citizen: observation. Acquisition context must be known to reuse the result.
      - 2nd class citizen: the result
      - => use O&M

- **Is the feature of interest a specimen?**
  - **no**
  - **yes**
    - **Non-Specimen**
      - Fol: Point
      - Fol: Curve
      - Fol: Surface
      - Fol: Solid
    - **Specimen**
      - Fixed point
      - Mobile – Curve

- **Is the result a coverage?**
  - **no**
  - **yes**
    - **Non-Coverage**
      - Single results
      - Multiple results in time
    - **Coverage**
      - Single results
      - Multiple results in time
6 O&M Extensions to INSPIRE

6.1 General

O&M is intended for cross-domain work and data exchange; it should mainly be used if provenance and quality of property values shall be provided with the data (e.g. to allow users to determine fitness-for-purpose). It is also useful for data discovery using the primary slots in the model (feature of interest, observed property, procedure) and can assist in data fusion across discipline boundaries. For intra-domain purposes, if the observation metadata is not important to the user, or the provider does not want to provide it, do not use O&M. Before a decision is reached to use O&M, the requirements must be clearly analysed to determine if an Observation-centric view is necessary or if a Result/Coverage-centric view will suffice. One key characteristic of O&M in this regard is the explicit relationships between the result and the feature of interest, sampling feature, procedure etc.

Figure 4: Observations Package

O&M is a very generic standard, allowing for very different design patterns depending on the domain as well as the Use Cases to be supported. Application of O&M within a technical community requires that the community agree on standard content for the key slots in the model, as well as on required extensions to the base classes provided within the standard. In particular, it is necessary to have standard vocabularies and extensions for the following:

i. The feature-types that may serve as the feature-of-interest (for direct observations) or for the sampled-feature (where a sampling feature is involved and described).
a. In the context of INSPIRE these feature types will often be drawn from the feature-concept-dictionary and associated feature-type-catalogue. When feature types are defined specifically to serve as the feature-of-interest or sampled-feature, these should be derived from the SF_SpatialSamplingFeature or SF_SpatialSamplingFeature Classes.

ii. The property definitions that may serve as the observed property. These should be assigned persistent identifiers (URIs) to allow them to be referenced in observation instances.

a. In the context of INSPIRE, the AbstractObservableProperty Class has been defined as a realisation of the metaclass GF_PropertyType. Various specialisations of the the AbstractObservableProperty Class have been provided as required in the Application Schema Observable Properties within the GCM

iii. The observation procedures used in the community. These may be expressed using SensorML or any suitable alternative formulation, and should be assigned persistent identifiers (URIs) to allow them to be referenced in observation instances.

a. The feature type Process, derived from the OM_Process feature type, has been provided within the Application Schema Processes within the GCM. The Process feature type provides the minimum requirements for the description of procedural information within INSPIRE. When using SensorML for the provision of this information, care must be taken to assure that all required concepts have been properly mapped to SensorML.

iv. The type of result to be provided for the observation. Vastly different encoding is required depending on the nature of the resulting data, which can range from a single value to a multidimensional result set.

a. In the context of INSPIRE, we have forgone the definition of specific result types. We provide recommendations as to which of the existing result types from both the various possible OGC encodings as well as other proprietary formats are best suited to specific purposes.

v. Specialisation of Observation classes by result type

a. In addition to the very generic OM_Observation, specialised observation types are provided by the O&M Standard. These have been further extended based on INSPIRE requirements. They are contained within the Application Schema Specialised Observations within the GCM.

vi. References to and from Observations

a. The Classes ObservationReference and ObservationSet, that can be used to reference an individual or a group of observations, have been provided. In addition, the Class EnvironmentalMonitoringFeatureReference allows for the OM_Observation to reference an EnvironmentalMonitoringFeature. These Classes are contained within the Application Schema Observation References within the GCM.

For each Application Schema, the Feature Catalogue has been provided in the relevant section. The following diagram gives an overview of the basic classes provided for the use of O&M in INSPIRE.

### 6.2 Feature of Interest

The Domain of an O&M Observation has the role featureOfInterest with respect to the OM_Observation. The featureOfInterest:

- is a thing from which the actual measurement is taken;
- may be a SF_SamplingFeature
- is not a device, but the actually focus of the observation.

Examples:
- A scene is a proxy for the land cover
- A citizen realizing an observation - the trajectory or exact location of the citizen making out the observation is the SamplingFeature -> not the citizen by itself
6.2.1 SamplingFeature vs. Domain Feature

In some cases, the featureOfInterest exists only because we have an observation, i.e. it is only defined in order to perform an observation of the real world; in this case, a specific sampling feature must be defined to serve as featureOfInterest. In other cases, a feature also used in other contexts within the domain will also serve as a featureOfInterest for an observation.

In the case where the featureOfInterest to be defined is a sampling feature, the base type for sampling features provided within the O&M specification, SF_SamplingFeature, should be used as the basis of derivation. The class SF_SpatialSamplingFeature has been derived from SF_SamplingFeature, and further specialised to support the requirements ensuing from features supporting one, two or three spatial dimensions. There are no temporally specialised SF_SamplingFeature because, for the time being, we cannot revisit time (contrary to space).

Recommendation 1 If the featureOfInterest of an observation is not a domain feature in its own right, then this feature should be (or be derived from) SF_SamplingFeature or its derived Types SF_SamplingPoint, SF_SamplingCurve, SF_SamplingSurface, SF_SamplingSolid or SF_Specimen.

NOTE See Figure 10 in ISO 19156:2011.

6.2.2 Sampling Features in SamplingCoverageObservations

Most of the specialised observations described in section “6.6.1 Specialised Observations” are derived from the specialized Observation class called SamplingCoverageObservation, which is defined in Annex D of ISO 19156:2011.

One of the constraints imposed on the SamplingCoverageObservation is that the featureOfInterest must be a SamplingFeature.

A further constraint imposed by use of the SamplingCoverageObservation is that the shape of the sampling feature of interest shall contain the spatial elements of the domain of the coverage result. The intention behind this constraint is explained in detail in Woolf, A. (ed.): Climate Science Modelling Language (CSML): Sampling Coverage Observations for the met/ocean domain (OGC 11-021):

“(A) spatial sampling feature may provide a summary description of a sampling geometry more fully specified in the result of a discrete coverage observation. In this sense it may be regarded as playing a role similar to the familiar ‘bounding box’ of dataset metadata (aiding discovery and spatial querying), but with a richer choice of geometries more suitable for observations. (…)

The shape of the spatial sampling feature may be regarded as a projection of the full spatiotemporal sampling geometry onto a spatial subspace. Projection of the full sampling geometry onto the temporal axis defines the time of the observation. Thus, the spatial sampling feature and observation phenomenon time broadly summarise the ‘where’ and ‘when’, respectively, of an observation, with the full geometric sampling complexity available in the domain of the coverage result.”

Thus, special care must be taken when defining SamplingFeatures for the use with SamplingCoverageObservations, to make sure that the domain of the result coverage is inside the geometry of the SamplingFeature.

Example: A SamplingFeature in the water domain is modelled as the grid used as the domain in the resulting coverage; this grid is defined covering the entirety of a specific lake. The SamplingFeature points to this specific lake as SampledFeature; the lake can in turn be sampled from the Terrestrial Hydrosphere.
6.2.3 SamplingFeature vs. SampledFeature

In many cases, only a featureOfInterest is provided in the form of a samplingFeature; a corresponding sampledFeature is often missing. The provision of standardized high-level sampledFeature referring to the media or realm covered is useful for providing a better understanding of the context of the featureOfInterest. Examples for high-level sampledFeatures are:

- Atlantic
- Atmosphere
- Danube River

While some of these high-level sampledFeatures may be INSPIRE feature types in their own right, e.g. Atlantic is a Sea (from SR), Danube is presumably a SurfaceWater feature (Hydro-Physical waters), it was not possible to identify an existing vocabulary that covers this type of concept across domains. The closest candidate identified is the Sweet Realms Ontology engineered by NASA (http://sweet.jpl.nasa.gov/2.1/). The following diagram shows the first level of classes in the Sweet Realms Ontology:

![Diagram showing the first level of classes in the Sweet Realms Ontology]

**Figure 5: High Level SampledFeature**

While this ontology covers most areas required, the internal structure is not always satisfactory to domain experts. If it suffices for a specific domain, we recommend its use; if it is not sufficient, we recommend the definition of an application specific sampled feature for this purpose. This sampledFeature should model a real-world geospatial feature that can aid non-domain experts in finding datasets of interest via textual and/or spatial search operations.

NOTE 1 The geometry of such a sampledFeature should intersect with the geometry of the corresponding samplingFeature, but the geometries need not be completely inside each other in either way.

**Recommendation 1** When using a samplingFeature as featureOfInterest, a domain feature should be provided as sampledFeature to provide the necessary context. One of these sampledFeatures should, if possible, be provided from a standardised vocabulary such as the Sweet Realms Ontology. In addition to these very abstract sampledFeatures, a more direct Sampled Feature should be provided that is of direct relevance to the samplingFeature.

NOTE 2 When the sampledFeature is tightly coupled with the measurement domain, it is possible to derive the sampledFeature from SF_SamplingFeature, and associate it with a high level sampledFeature from a standardised vocabulary.
6.3 Observed Property

The observedProperty of an OM_Observation is the property of the featureOfInterest that is observed during the act of observation. For example, if the temperature of the featureOfInterest is measured, then the observedProperty is 'temperature'.

The observedProperty in O&M must be a reference to some definition of that property. So typically this would be to an item in a published vocabulary. However, it is quite common in practice that definitions of observed properties in published vocabularies are not specific enough to allow end users to interpret exactly which property was observed.

For example, the observedProperty may be 'radiance', and this may be defined in a vocabulary. However the actual property observed is radiance at a particular wavelength or wavelength range e.g. between 300-400 nm. Another example is temperature. It may not be sufficient to simply state that an Observation has the observedProperty 'temperature'. It may be important to know that it is 'Daily Maximum Temperature'.

INSPIRE has developed an ObservableProperty model (Figure 5) to provide a framework for extending a pre-defined term in a vocabulary with additional information, such as constraints (e.g. the earlier wavelength example), or statistical measures (e.g. the earlier temperature example).

<table>
<thead>
<tr>
<th>Examples:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water: The SamplingFeature points to a specific River as sampledFeature; the River can in turn be sampled from the &quot;Terrestrial Hydrosphere&quot;.</td>
</tr>
<tr>
<td>Air Quality: The SamplingFeature that is featureOfInterest of an observation is the air bubble around the air intake of an air quality monitoring station, the sampledFeature is the representative area around the station. This representative area can be derived from SF_SamplingFeature, and in turn reference the atmosphere as its sampledFeature.</td>
</tr>
</tbody>
</table>
6.3.1 AbstractObservableProperty

The AbstractObservableProperty class is the root of the ObservableProperty model. It is implemented by two specialisations: ObservableProperty and CompositeObservableProperty.

6.3.2 ObservableProperty

At its simplest an ObservableProperty simply carries a reference to a phenomenon definition in a codelist with optional units of measure. However an ObservableProperty definition may be augmented using Constraints and/or Statistical Measures to create a more full definition of the observed property.
6.3.3 Statistical Measures

The StatisticalMeasure is used to describe some statistical grouping of the data. It has an attribute where one can provide the statistical function being applied (i.e. mean). In addition, it provides attributes for the description of what the statistics are being applied to, i.e. over what dimension the mean is being taken over.

For the provision of hourly means, one would enter a reference to "mean" in the statisticalFunction attribute and then provide a duration of an hour in the aggregationTimePeriod attribute. StatisticalMeasures can be aggregated over time, spatial dimensions or any other defined aggregation type. An example of a spatial aggregation would be ‘maximum per km²’.

Note that a StatisticalMeasure may be derived from another StatisticalMeasure. For example Mean Monthly Maximum Temperature derived from Mean Daily Maximum Temperature.

6.3.4 Constraints

In order to provide other constraints on ObservableProperties, the type Constraint has been created and can be associated with an ObservableProperty. Constraint types have been provided for scalar, range and category constraints, as well as a generic OtherConstraint data type. The following list provides examples for the constraint types defined:

- Scalar Constraint: A numerical scalar constraint on some property e.g. length >= 1m
- Range Constraint: A numerical range constraint on some property e.g. wavelength >=300nm and wavelength <=600nm. To be used when data is observed in particular bands or groupings based on a numerical quantity.
- Category Constraint: A constraint based on some qualifying category. e.g. colour = ‘Red’. The value (‘Red’) of the constraint (‘colour’) can be any string, although it may be desirable to constrain this in particular application domains.
- Other Constraint: Any other constraint type not easily expressed using the other Constraint types. This type may be specialised or the simple description attribute may be used to provide a free text description of the constraint.

6.3.5 CompositeObservableProperty

Usually, when performing multiple observations on one featureOfInterest, one provides a separate ObservableProperty element for each Phenomenon being observed. However, in certain cases where either a) there is a strong link between the Phenomena or b) the multiple phenomena are clearly observed as part of the same Observation, these Phenomena may be provided together in one Observation. In this case a CompositeObservableProperty can be defined that groups together multiple Phenomena (ObservableProperty) into one CompositeObservableProperty element.

An example of a strongly linked pair of Phenomena is wind speed and wind direction.

6.3.6 Feature catalogue

Feature catalogue metadata

<table>
<thead>
<tr>
<th>Application Schema</th>
<th>INSPIRE Application Schema Observable Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version number</td>
<td>2.0rc3</td>
</tr>
</tbody>
</table>

Types defined in the feature catalogue

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Stereotypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CategoryConstraint</td>
<td>Observable Properties</td>
<td>«dataType»</td>
</tr>
<tr>
<td>Constraint</td>
<td>Observable Properties</td>
<td>«dataType»</td>
</tr>
<tr>
<td>OtherConstraint</td>
<td>Observable Properties</td>
<td>«dataType»</td>
</tr>
<tr>
<td>RangeBounds</td>
<td>Observable Properties</td>
<td>«dataType»</td>
</tr>
<tr>
<td>RangeConstraint</td>
<td>Observable Properties</td>
<td>«dataType»</td>
</tr>
</tbody>
</table>
Type | Package | Stereotypes
--- | --- | ---
ScalarConstraint | Observable Properties | «dataType»

6.3.6.1. Data types

6.3.6.1.1. CategoryConstraint

CategoryConstraint

Name: Category Constraint
Subtype of: Constraint
Definition: A constraint based on some qualifying category. e.g. colour = 'Red'.
Description: A constraint based on some qualifying category. e.g. colour = 'Red'. The value ('Red') of the constraint ('colour') can be any string, although it may be desirable to constrain this in particular application domains.
Stereotypes: «dataType»

 Attribute: comparison

Name: comparison
Value type: ComparisonOperatorValue
Definition: A comparison operator. In the case of a category constraint it should be 'equalTo' or 'notEqualTo'.
Multiplicity: 1

 Attribute: value

Name: value
Value type: CharacterString
Definition: The value of the property that is constrained e.g. 'blue' (if the constrained property is colour)
Multiplicity: 1..*

6.3.6.1.2. Constraint

Constraint

Name: Constraint
Definition: A constraint on some property e.g. wavelength = 200nm.
Description: A constraint on some property e.g. wavelength = 200nm. This property is typically not the same property as the base phenomenon of the observed property. e.g. the observed property has a base phenomenon 'radiance'. a constraint is added to say 'wavelength = 200nm' So the overall ObservableProperty which is represented is 'radiance where wavelength = 200nm'
The Constraint class is specialised into several specific classes covering Scalar, Range and Categorical constraints
Stereotypes: «dataType»

 Attribute: constrainedProperty

Name: constrainedProperty
Value type: PhenomenonTypeValue
Definition: The property being constrained. e.g. 'colour' if the constraint is 'colour = blue'
Multiplicity: 0..1

 Attribute: label

Name: description
Value type: CharacterString
Definition: A human readable title for the constraint as a whole
Multiplicity: 0..1
### 6.3.6.1.3. *OtherConstraint*

<table>
<thead>
<tr>
<th><strong>Name:</strong></th>
<th>Other Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subtype of:</strong></td>
<td>Constraint</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>A constraint, not modelled in a structured way, but may be described using the freetext 'description' attribute.</td>
</tr>
<tr>
<td><strong>Stereotypes:</strong></td>
<td>«dataType»</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Attribute:</strong></th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>description</td>
</tr>
<tr>
<td><strong>Value type:</strong></td>
<td>CharacterString</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>A description of the constraint.</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

### 6.3.6.1.4. *RangeBounds*

<table>
<thead>
<tr>
<th><strong>Name:</strong></th>
<th>Range Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition:</strong></td>
<td>The start and end bounding values of a numerical range (e.g. start &gt;=50, end &lt;=99)</td>
</tr>
<tr>
<td><strong>Stereotypes:</strong></td>
<td>«dataType»</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Attribute:</strong></th>
<th>startComparison</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>startComparison</td>
</tr>
<tr>
<td><strong>Value type:</strong></td>
<td>ComparisonOperatorValue</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>The comparator used for the lower range limit (e.g. greaterThanOrEqualTo)</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Attribute:</strong></th>
<th>rangeStart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>rangeStart</td>
</tr>
<tr>
<td><strong>Value type:</strong></td>
<td>Real</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>The lower limit of the range.</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Attribute:</strong></th>
<th>endComparison</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>endComparison</td>
</tr>
<tr>
<td><strong>Value type:</strong></td>
<td>ComparisonOperatorValue</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>The comparator used for the upper range limit (e.g. lessThan)</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Attribute:</strong></th>
<th>rangeEnd</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>rangeEnd</td>
</tr>
<tr>
<td><strong>Value type:</strong></td>
<td>Real</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>The upper limit of the range.</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

### 6.3.6.1.5. *RangeConstraint*

<table>
<thead>
<tr>
<th><strong>Name:</strong></th>
<th>Range Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subtype of:</strong></td>
<td>Constraint</td>
</tr>
<tr>
<td><strong>Definition:</strong></td>
<td>A numerical range constraint on some property e.g. wavelength &gt;=300nm and wavelength &lt;=600nm</td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>A numerical range constraint on some property e.g. wavelength &gt;=300nm and wavelength &lt;=600nm e.g. To be used when data is observed in particular bands or groupings based on a numerical quantity.</td>
</tr>
</tbody>
</table>
### RangeConstraint

**Stereotypes:** «dataType»

<table>
<thead>
<tr>
<th>Attribute: value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> value</td>
</tr>
<tr>
<td><strong>Value type:</strong> RangeBounds</td>
</tr>
<tr>
<td><strong>Definition:</strong> The numerical value range of the property that is constrained</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong> 1..*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute: uom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> uom</td>
</tr>
<tr>
<td><strong>Value type:</strong> UnitOfMeasure</td>
</tr>
<tr>
<td><strong>Definition:</strong> Units of measure used in the constraint</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong> 0..1</td>
</tr>
</tbody>
</table>

### 6.3.6.1.6. ScalarConstraint

**ScalarConstraint**

<table>
<thead>
<tr>
<th>Attribute: value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> value</td>
</tr>
<tr>
<td><strong>Value type:</strong> Real</td>
</tr>
<tr>
<td><strong>Definition:</strong> The numerical value of the property that is constrained</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong> 1..*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute: comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> comparison</td>
</tr>
<tr>
<td><strong>Value type:</strong> ComparisonOperatorValue</td>
</tr>
<tr>
<td><strong>Definition:</strong> The comparator to be used in the constraint e.g. greaterThan</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong> 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute: uom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> uom</td>
</tr>
<tr>
<td><strong>Value type:</strong> UnitOfMeasure</td>
</tr>
<tr>
<td><strong>Definition:</strong> Units of measure used in the constraint</td>
</tr>
<tr>
<td><strong>Multiplicity:</strong> 0..1</td>
</tr>
</tbody>
</table>

### 6.3.6.1.7. StatisticalMeasure

**StatisticalMeasure**

<table>
<thead>
<tr>
<th>Attribute: label</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> label</td>
</tr>
<tr>
<td><strong>Value type:</strong> CharacterString</td>
</tr>
</tbody>
</table>
### StatisticalMeasure

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>statisticalFunction</td>
<td>A statistical function e.g. (mean)</td>
<td>0..1</td>
</tr>
</tbody>
</table>

### Attribute: statisticalFunction

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregationTimePeriod</td>
<td>A temporal range over which a statistic is calculated. e.g. A day, An hour.</td>
<td>0..1</td>
</tr>
</tbody>
</table>

### Attribute: aggregationLength

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregationArea</td>
<td>A two dimensional spatial range over which a statistic is calculated for example 1 square metre</td>
<td>0..1</td>
</tr>
</tbody>
</table>

### Attribute: aggregationVolume

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>otherAggregation</td>
<td>Any other type of aggregation.</td>
<td>0..1</td>
</tr>
</tbody>
</table>

### Association role: derivedFrom

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComparisonOperatorValue</td>
<td>One statistical measure may be derived from another. e.g. Monthly Maximum temperatures may be derived from Daily Mean temperatures.</td>
<td>0..1</td>
</tr>
</tbody>
</table>

### 6.3.6.2. Enumerations

#### 6.3.6.2.1. ComparisonOperatorValue

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>derived from</td>
<td>One statistical measure may be derived from another. e.g. Monthly Maximum temperatures may be derived from Daily Mean temperatures.</td>
<td>0..1</td>
</tr>
</tbody>
</table>
ComparisonOperatorValue

| Definition                                | URI: | Value: equalTo
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>An enumeration of comparison operators</td>
<td></td>
<td>Exactly equal to</td>
</tr>
</tbody>
</table>
| URI:                                      |      | Value: notEqualTo
|                                            |      | Not exactly equal to |
|                                            |      | Value: lessThan
|                                            |      | Less than |
|                                            |      | Value: greaterThan
|                                            |      | Greater Than |
|                                            |      | Value: lessThanOrEqualTo
|                                            |      | Less than or exactly equal to |
|                                            |      | Value: greaterThanOrEqualTo
|                                            |      | Greater than or exactly equal to |

6.3.6.3. Code lists

6.3.6.3.1. PhenomenonTypeValue

<table>
<thead>
<tr>
<th>Name: Phenomenon Type Value</th>
<th>Definition: A code list of phenomena (e.g. temperature, wind speed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>A code list of phenomena. This code list itself is an empty placeholder and should be extended and specified for any thematic domain.</td>
</tr>
<tr>
<td>Extensibility:</td>
<td>open</td>
</tr>
</tbody>
</table>

The allowed values for this code list comprise the values of the following code lists and additional values at any level defined by data providers:

- CFStandardNamesValue (INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5])
- ProfileElementParameterNameValue (INSPIRE Data specification on Soil [DS-D2.8.III.3])
- SoilDerivedObjectParameterNameValue (INSPIRE Data specification on Soil [DS-D2.8.III.3])
- SoilProfileParameterNameValue (INSPIRE Data specification on Soil [DS-D2.8.III.3])
- SoilSiteParameterNameValue (INSPIRE Data specification on Soil [DS-D2.8.III.3])
- EU_AirQualityReferenceComponentValue (INSPIRE Data specification on Atmospheric Conditions and Meteorological Geographical Features [DS-D2.8.III.13-14])
- BODC_P01ParameterUsageValue (INSPIRE Data specification on Oceanographic Geographical Features [DS-D2.8.III.15])
### 6.3.6.3.2. StatisticalFunctionTypeValue

<table>
<thead>
<tr>
<th>Name:</th>
<th>Statistical Function Type Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>A code list of statistical functions (e.g. maximum, minimum, mean)</td>
</tr>
<tr>
<td>Description:</td>
<td>A code list of statistical functions. This code list itself is an empty placeholder and should be extended and specified for any thematic domain.</td>
</tr>
<tr>
<td>Extensibility:</td>
<td>any</td>
</tr>
<tr>
<td>Identifier:</td>
<td></td>
</tr>
<tr>
<td>Values:</td>
<td>The allowed values for this code list comprise any values defined by data providers.</td>
</tr>
</tbody>
</table>

### 6.3.6.4. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

#### 6.3.6.4.1. Any

<table>
<thead>
<tr>
<th>Any</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.2. Area

<table>
<thead>
<tr>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.3. BODC_P01ParameterUsageValue

<table>
<thead>
<tr>
<th>BODC_P01ParameterUsageValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
<tr>
<td>Reference:</td>
</tr>
<tr>
<td>Definition:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.4. CFStandardNamesValue

<table>
<thead>
<tr>
<th>CFStandardNamesValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.5. CharacterString

<table>
<thead>
<tr>
<th>CharacterString</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.6. EU_AirQualityReferenceComponentValue

<table>
<thead>
<tr>
<th>EU_AirQualityReferenceComponentValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
<tr>
<td>Reference:</td>
</tr>
<tr>
<td>Definition:</td>
</tr>
</tbody>
</table>

#### 6.3.6.4.7. Length

<table>
<thead>
<tr>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package:</td>
</tr>
</tbody>
</table>
6.3.6.4.8. **ProfileElementParameterNameValue**

**ProfileElementParameterNameValue**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Soil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference:</td>
<td>INSPIRE Data specification on Soil [DS-D2.8.III.3]</td>
</tr>
<tr>
<td>Definition:</td>
<td>list of properties that can be observed to characterize the profile element. The allowed values for this code list comprise a number of pre-defined values and narrower values defined by data providers. This code list is hierarchical.</td>
</tr>
<tr>
<td>Description:</td>
<td>Basically these parameters can be divided in several major groups like:</td>
</tr>
<tr>
<td></td>
<td>• Chemical parameters</td>
</tr>
<tr>
<td></td>
<td>• Physical parameters</td>
</tr>
<tr>
<td></td>
<td>• Biological parameters</td>
</tr>
</tbody>
</table>

6.3.6.4.9. **Real**

**Real**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Numerics</th>
</tr>
</thead>
</table>

6.3.6.4.10. **SoilDerivedObjectParameterNameValue**

**SoilDerivedObjectParameterNameValue**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Soil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference:</td>
<td>INSPIRE Data specification on Soil [DS-D2.8.III.3]</td>
</tr>
<tr>
<td>Definition:</td>
<td>list of soil related properties that can be derived from soil and other data. The allowed values for this code list comprise a number of pre-defined values and narrower values defined by data providers. This code list is hierarchical.</td>
</tr>
<tr>
<td>Description:</td>
<td>Basically these parameters can be divided in several major groups like:</td>
</tr>
<tr>
<td></td>
<td>• Chemical parameters</td>
</tr>
<tr>
<td></td>
<td>• Physical parameters</td>
</tr>
<tr>
<td></td>
<td>• Biological parameters</td>
</tr>
</tbody>
</table>

6.3.6.4.11. **SoilProfileParameterNameValue**

**SoilProfileParameterNameValue**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Soil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference:</td>
<td>INSPIRE Data specification on Soil [DS-D2.8.III.3]</td>
</tr>
<tr>
<td>Definition:</td>
<td>list of properties that can be observed to characterize the soil profile. The allowed values for this code list comprise a number of pre-defined values and narrower values defined by data providers. This code list is hierarchical.</td>
</tr>
<tr>
<td>Description:</td>
<td>Basically these parameters can be divided in several major groups like:</td>
</tr>
<tr>
<td></td>
<td>• Chemical parameters</td>
</tr>
<tr>
<td></td>
<td>• Physical parameters</td>
</tr>
<tr>
<td></td>
<td>• Biological parameters</td>
</tr>
</tbody>
</table>

6.3.6.4.12. **SoilSiteParameterNameValue**

**SoilSiteParameterNameValue**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Soil</th>
</tr>
</thead>
</table>
SoilSiteParameterNameValue

Reference: INSPIRE Data specification on Soil [DS-D2.8.III.3]
Definition: List of properties that can be observed to characterize the soil site. The allowed values for this code list comprise a number of pre-defined values and narrower values defined by data providers.
Description: Basically these parameters can be divided in several major groups like:

- Chemical parameters
- Physical parameters
- Biological parameters

6.3.6.4.13. TM_Duration

TM_Duration

Package: Temporal Objects

6.3.6.4.14. UnitOfMeasure

UnitOfMeasure (abstract)

Package: Units of Measure

6.3.6.4.15. Volume

Volume

Package: Units of Measure

6.4 Process

Within O&M, the base definition of OM_Process is an empty class. For use within INSPIRE, the following 2 options have been determined:

- SensorML: Using SensorML has the advantage of maintaining compliance with the wider SWE scope. To bring this closer to INSPIRE, we propose a standardized mapping of agreed upon attributes to the SensorML model. It is important to keep in mind that SensorML was developed by a team whose main experience was remote sensing, so it may not be suitable for all domains. Much will depend on how SensorML V2.0 evolves.
- Process: The Process class defined within INSPIRE allows for the lightweight provision of procedural information. The disadvantage is that it goes away from the base standard and thus must be optional to allow for re-use of domain standards

The Class Process has been created for this purpose.

IR Requirement
Annex I, Section 7.5
Requirements for Observations

Where the OM_Observation type or any sub-type thereof is used to make data available, the following requirements shall apply:

1. The Process type shall be used to indicate the procedure used in an OM_Observation.
6.4.1 Process Parameters

The Process Parameters define either instrumentation settings for a specific measurement or measurement series, or other information, sometimes stemming from other observations or measurements, which are of relevance to interpreting this observation. The Process Parameters show which types of additional information are to be expected within the observation, the actual values detailing this information are then stored in the parameter attribute of the observation. This parameter attribute should use the value defined in the Process Parameters as the key, the value can be a single field, but also a complex structure (question how to constrain the type of structure allowed). This allows for the definition of a re-usable process that can be further configured through the Process Parameters.
IR Requirement
Annex I, Section 7.5
Requirements for Observations

Where the OM_Observation type or any sub-type thereof is used to make data available, the following requirements shall apply:

(...)

(4) If the processParameter attribute is present in the procedure property of an OM_Observation object, its value (a name) shall be included in the parameter attribute of the OM_Observation object.

6.4.1.1. Types of Process Parameters expected

Very different types of information will be supplied via the Process Parameters. At present, we have identified the following categories:

- Results of related "observations": not necessarily available in the form of an observation, the information provided are the results of related observations relevant to the current observation. Examples of this are:
  - the water hardness of the river being sampled from
  - the width of the river at the measurement point
- Temporal information not currently covered by the OM_Observation. This is often necessary in forecasts and modeling. Examples of this are:
  - Analysis Time
  - Forecast period
- Instrument settings can be stored via process parameters to allow for the reuse of this process instance in various settings. Examples of this are:
  - sampling rate
  - minimum & maximum offset
- EnsembleMember can be specified as an easy way of providing data from aggregate sensors. In this case, the Process Parameters provide information in which exact sensor was used.

6.4.1.2. Referencing Process Parameters

For true interoperability, the re-use of Process Parameters stemming from an external source (vocabulary) is necessary. Otherwise, it will often not be clear what type of information is to be stored here. Unfortunately, this will not always be possible as in most cases these vocabularies do not yet exist, and where they do, we cannot be sure that they will encompass all requirements.

EXAMPLE The following example shows how the Process Parameters stored under the referenced procedure link to the observation parameters (note that the URIs have been shortened for readability, "inspire.jrc.ec.europa.eu" has been reduced to "inspire").

Process
- identifier: ukmo_global_model
- processParameter: http://inspire/processParameterValue.html#AnalysisTime
- processParameter: http://inspire/processParameterValue.html#AssimilationWindow

OM_Observation
- phenomenonTime: 00z 15/05/2011 - 00z 21/05/2011
- resultTime: 0420z 15/05/2011
- parameter: Name: http://inspire/processParameterValue.html#AnalysisTime Value: 00z 15/05/2011
6.4.1.3. Constraints for Process Parameters

In order to assure consistency between the Process Parameters in the procedure and the parameters within the observation, constraints should be applied. At the same time, we do not wish to specialise the OM_Observation to maintain compatibility with thematic O&M based standards. Thus, we will not formally provide constraints within the OM_Observation, but recommend the provision of Schematron rules together with the schema in order to assure that all keys used in the observation parameters must also be listed under the process parameters.

6.4.2 Process Encoding with SensorML

<table>
<thead>
<tr>
<th>INSPIRE Attribute &amp; Datatype</th>
<th>SensorML XPath</th>
</tr>
</thead>
<tbody>
<tr>
<td>documentation : DocumentCitation</td>
<td>.../sml:documentation/sml:Document/gml:description</td>
</tr>
<tr>
<td>shortName</td>
<td>NA</td>
</tr>
<tr>
<td>date</td>
<td>.../sml:documentation/sml:Document/sml:date</td>
</tr>
<tr>
<td>link</td>
<td>.../sml:documentation/sml:Document/sml:onlineResource/@xlink:href</td>
</tr>
<tr>
<td>inspireId : Identifier</td>
<td>NOTE The inspireId should be provided in an individual IdentifierList entry. The identifier should carry the name inspireId. .../sml:identification[1]/sml:IdentifierList/sml:identifier/@name = inspireId</td>
</tr>
<tr>
<td>localId</td>
<td>.../sml:identification/sml:IdentifierList/sml:identifier/sml:Term/sml:value</td>
</tr>
<tr>
<td>versionId</td>
<td>NOTE The pair of name and description for one process parameter must always be grouped together in one ClassifierList</td>
</tr>
<tr>
<td>name : CharacterString</td>
<td>.../sml:identification/sml:IdentifierList/sml:identifier/sml:Term/sml:value</td>
</tr>
<tr>
<td>.../sml:identification/sml:IdentifierList/sml:identifier/@name = procedureName</td>
<td></td>
</tr>
<tr>
<td>processParameter : ProcessParameter</td>
<td>NOTE The pair of name and description for one process parameter must always be grouped together in one ClassifierList</td>
</tr>
<tr>
<td>description</td>
<td>.../sml:classification/sml:ClassifierList/sml:classifier/sml:Term/sml:value</td>
</tr>
<tr>
<td>name</td>
<td>.../sml:classification/sml:ClassifierList/sml:classifier/sml:Term/sml:value</td>
</tr>
<tr>
<td>responsibleParty : RelatedParty</td>
<td>.../sml:contact/sml:ResponsibleParty</td>
</tr>
<tr>
<td>type : CharacterString</td>
<td>.../sml:classification/sml:ClassifierList/sml:classifier/sml:Term/sml:value</td>
</tr>
</tbody>
</table>

NOTE: All XPaths should start with “/sml:SensorML/sml:member/sml:Component”, for brevity this has been reduced to “…”
6.4.3 Feature catalogue

Feature catalogue metadata

<table>
<thead>
<tr>
<th>Application Schema</th>
<th>INSPIRE Application Schema Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version number</td>
<td>2.0rc3</td>
</tr>
</tbody>
</table>

Types defined in the feature catalogue

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Stereotypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Processes</td>
<td>«featureType»</td>
</tr>
<tr>
<td>ProcessParameter</td>
<td>Processes</td>
<td>«dataType»</td>
</tr>
</tbody>
</table>

6.4.3.1 Spatial object types

6.4.3.1.1 Process

Process

<table>
<thead>
<tr>
<th>Name: Process</th>
<th>Subtype of: OM_Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Description of an observation process</td>
</tr>
<tr>
<td>Stereotypes:</td>
<td>«featureType»</td>
</tr>
</tbody>
</table>

Attribute: documentation

<table>
<thead>
<tr>
<th>Name: documentation</th>
<th>Value type: DocumentCitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Further information (online/offline) associated with the process.</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>0..*</td>
</tr>
<tr>
<td>Stereotypes:</td>
<td>«voidable»</td>
</tr>
</tbody>
</table>

Attribute: inspireld

<table>
<thead>
<tr>
<th>Name: inspireld</th>
<th>Value type: Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>External object identifier of the spatial object.</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>1</td>
</tr>
<tr>
<td>Stereotypes:</td>
<td>«voidable»</td>
</tr>
</tbody>
</table>

Attribute: name

<table>
<thead>
<tr>
<th>Name: name</th>
<th>Value type: CharacterString</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Name of the Process</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>0..1</td>
</tr>
<tr>
<td>Stereotypes:</td>
<td>«voidable»</td>
</tr>
</tbody>
</table>

Attribute: processParameter

<table>
<thead>
<tr>
<th>Name: process parameter</th>
<th>Value type: ProcessParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Parameter controlling the application of the process and as a consequence its output.</td>
</tr>
<tr>
<td>Description:</td>
<td>Typical examples of using processParameter are: description of instrumentation settings for a specific measurement or measurement series; description of initial conditions in numerical computations e.g. simulations.</td>
</tr>
</tbody>
</table>

NOTE The values of a processParameter are stored in OM_Observation.parameter>NamedValue.value as they are specific to the event of the observation and not to the applied process. The relevant OM_Observation.parameter>NamedValue.name shall be the same with Process.processParameter>ProcessParameter.name.
Process

EXAMPLE Analysis time of a forecast

- Instance of Process

Process.processParameter->ProcessParameter.name:

- Instance of OM_Observation

OM_OObservation.parameter>NamedValue.name:
OM_Observation.parameter>NamedValue.value:
00z-15/05/2011

Multiplicity: 0..*
Stereotypes: «voidable»

Attribute: responsibleParty

Name: responsible party
Value type: RelatedParty
Definition: Individual or organisation related to the process.
Description: EXAMPLE For a numerical simulation a responsible party may be the NWP centre which conducted the simulation
Multiplicity: 1..*
Stereotypes: «voidable»

Attribute: type

Name: type
Value type: CharacterString
Definition: Type of process.
Description: EXAMPLE raingauge, numerical model.
Multiplicity: 1
Stereotypes: «voidable»

6.4.3.2. Data types

6.4.3.2.1. ProcessParameter

ProcessParameter

Name: Process Parameter
Definition: Description of the given parameter
Stereotypes: «dataType»

Attribute: description

Name: description
Value type: CharacterString
Definition: Description of the process parameter.
Multiplicity: 0..1

Attribute: name

Name: name
Value type: ProcessParameterNameValue
Definition: Name of the process parameter.
Multiplicity: 1
Values: The allowed values for this code list comprise any values defined by data providers.
6.4.3.3. Code lists

6.4.3.3.1. ProcessParameterNameValue

| Name:          | Process Parameter Name Value |
| Definition:    | A code list of names of process parameters. |
| Description:   | This code list itself is an empty placeholder and should be extended and specified for any thematic domain. |
| Extensibility: | any |
| Identifier:    | |
| Values:        | The allowed values for this code list comprise any values defined by data providers. |

6.4.3.4. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

6.4.3.4.1. CharacterString

| Package:     | Text |

6.4.3.4.2. DocumentCitation

| Package:     | Base Types 2 |
| Reference:   | INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5] |
| Definition:  | Citation for the purposes of unambiguously referencing a document. |

6.4.3.4.3. Identifier

| Package:     | Base Types |
| Reference:   | INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5] |
| Definition:  | External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object. |
| Description: | NOTE1 External object identifiers are distinct from thematic object identifiers. |

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTE 2</td>
<td>The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTE 3</td>
<td>The unique identifier will not change during the life-time of a spatial object.</td>
</tr>
</tbody>
</table>

6.4.3.4.4. RelatedParty

| Package:     | Base Types 2 |
| Reference:   | INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5] |
| Definition:  | An organisation or a person with a role related to a resource. |
| Description: | NOTE 1 A party, typically an individual person, acting as a general point of contact for a resource can be specified without providing any particular role. |
6.5 Result

In many cases, it will be possible to provide the result data using either GML coverages or the result types provided in SWE Common. These allow for the standardized encoding of a wide range of simple and complex result types including spatial and temporal coverages.

The following examples from ISO 19156:2011 illustrate different types of observations with various types of spatial sampling features. They are listed in order to provide help in understanding the different types of results that would be consistent with the spatial sampling feature type in these observations:

<table>
<thead>
<tr>
<th>Observation class</th>
<th>Example</th>
<th>Spatial sampling feature</th>
<th>Coverage result</th>
</tr>
</thead>
</table>
| Profile           | Expendable bathythermograph observation of seawater temperature | SF_SamplingCurve | • one-dimensional grid at fixed (x,y,t) within four-dimensional (x-y-z-t) CRS  
• grid axis aligned with CRS z-axis |
| ProfileTimeSeries | Radar wind profiler measurement | SF_SamplingCurve | • two-dimensional grid at fixed (x,y) within four-dimensional (x,y,z,t) CRS  
• grid axes aligned with CRS z- and t-axes |
| Trajectory        | Pollutant concentration from mobile air quality sensor | SF_SamplingCurve | • one-dimensional grid within four-dimensional (x-y-z-t) CRS |
| xSection          | Vertical profiles of water current measurements taken by an acoustic doppler current profiler towed along a ship's track | SF_SamplingSurface | • two-dimensional grid within four-dimensional (x-y-z-t) CRS  
• one grid axis aligned with CRS z-axis |
| GridTimeSeries    | Time-series of 3-D velocity field from a finite-difference seismic model | SF_SamplingSolid | • four-dimensional grid within four-dimensional (x-y-z-t) CRS |

The result should contain validation information, either on general result level or in coverages this could also be on specific element level.

Recommendation 2: Individual observations should be provided per individual Phenomenon. While it is possible to provide multiple components using a CompositeObservableProperty, this should only be done if there is a strong link between the Phenomena, to allow for easier access to an individual Phenomenon.

6.5.1 State of the Art

At present, there is no clear consensus on State of the Art across thematic domains on result encoding. While the OGC Sensor Web community foresees the use of the encoding types provided by the OGC SWE Common package, thematic communities have chosen alternative result encoding options. Examples of this are:

- WaterML encoding
- GML Features
- GML Coverages
- NetCDF and other proprietary encodings

Further discussion and prototyping would be required, both within the INSPIRE community but perhaps also in collaboration with OGC, in order to determine what result encoding option or options are best suited to the needs of INSPIRE.

### 6.5.2 SWE Result Encoding

For the encoding of observations on points, be they individual one-time observations or time series, the results types provided in the OGC SWE Common package should suffice for most INSPIRE requirements.

For individual one-time observations, the following simple result types are foreseen:

![Diagram of Simple Components](image)

**Figure 8: Results - Simple Components**

In addition, specific range results types for results expressed a range are provided. These pertain to range results of type Quantity, Count, Category and Time, and are illustrated in the following diagram:
Figure 9: Results - Range Components

In addition to providing the results on individual one-time observations, we must also be able to provide time-series results, so the results of multiple measurements over time on one Feature-of-Interest. For this purpose, we recommend the use of the DataArray Type.

This type allows for the provision of results consisting of multiple elements, so an individual result entry could consist of the following 4 elements:

- Timestamp: the time the individual measurement was made
- Validity: a flag showing the validity of the individual measurement
- Verification Status: a flag showing the verification status, as in some cases preliminarily verified data will be made available, for example in the provision of near-real-time data.
- Measurement Result: the actual value coming from the measurement.

When using the DataArray type with such a multipart result, these 4 fields would be described in the elementType attribute; the encoding of these elements is provided in the encoding attribute, while the encoded timeseries is provided in the values attribute.
6.5.3 GML Coverage Result Encoding

In the Result/Coverage-centric view, the result (generally a coverage in this case) is the primary object of interest while the description of the observation process is just metadata of the result; coverages are “first class citizens” and observation is metadata about the coverage.

In this context, it is possible to envision design patterns that forgo provision of procedural metadata entirely as this in not of further relevance for the interpretation of the result. However, there are also cases where the procedural metadata can be of relevance to the further processing chain. In such cases, the coverage metadata attribute can be used to supply such information.

The following datatypes have been identified as suitable for the provision of observational metadata to coverages:

- MI_Metadata: acquisition source
- OM_Observation
- ObservingCapability

6.5.4 Out-of-Band Result Encoding

Various INSPIRE themes (GE, AC, …) have mentioned the requirement for out-of-band result encoding, especially for the provision of large result sets. This is also necessary for the use of industry standards for result encoding (i.e. NetCDF, SEG-Y, LAS, WITSML). While it would be possible to reference resources encoded in these industry standards directly from the observation, it does not provide the user with the information required to determine if results of this type can be further processed on the available systems.

The actual encoding to be used for Out-of-Band results is still under discussion. A discussion paper has been provided as an annex to this document. The following options are presented in the discussion paper and summarized here:

- Option 1 (in-band): Embed the result using GML Coverage encoding: straightforward coverage result provided with the observation (so only in-band option)

- Option 2: (out-of-band) XLink the entire om:result: worst case – result is just an xlink, no information on the type of result or access modes

- Option 3 (out-of-band): Define an INSPIRE specific referencing “out-of-band” result type: almost worst case – does provide additional info on alternative access methods, but no further information on the result type

- Option 4 (out-of-band): Link to an external rangeSet using gml:File based on CSML storage description, provides basic information inline describing an external resource. The following problems were discussed:
  - inspire_om:ExternalStorageDescriptor should maybe be modified so as not to encode result type specific elements, currently there are specific elements by result type. Problem is that result types vary greatly, so difficult to provide a generic descriptor.
  - how to deal with multiple result files (only one gml:file element is allowed under gml:rangeSet)
  - No alternative access methods (see 5)

- Option 5 (out-of-band): gml:File with AlternativeEncodings: Basically a merge of options 3&4, so access methods plus description of the external resource
• Option 6 (out-of-band): Using SWE Common DataStream & BinaryEncoding: SWE model. Uses the swe:DataArray type for description of the data set, with further encoding information in the swe:BinaryEncoding element. Values are then externalized through xlink. Problem is that doesn’t seem suited for coverages
  o One idea that this option brought up was to what extent one could use the swe:DataArray type instead of the inspire_om:ExternalStorageDescriptor – to be checked

• Option 7 CSML Pattern (New, so not yet in discussion paper): use CSML encoding style – in the gml:rangeSet the gml:valueComponent is xlinked to an AggregatedArray element which in turn links to the external resource. It’s a bit of an ugly workaround, so needs to be further thought through.

6.6 Observation

In order to maintain compatibility with the various domain specific standards based on O&M such as CSML or WaterML 2.0, we have decided not to specialise OM_Observation with additional attributes within INSPIRE; all specialisations provided only pertain to the addition of constraints. At the same time, we do see a requirement to provide further information on the observational process within the observation itself. For example, it would often be valuable to store a limited amount of calibration information linked to the observation instance and not the procedure.

For this purpose, we propose the use of the observation “parameter” attribute. This attribute allows for provision of diverse types of data as key-value pairs. As most of the additional observation-relevant information identified is related to the procedure used during the observation, which is defined in the process, the processParameter attribute has been provided in the Process type to describe process runtime parameters. These should be clearly documented in the process description, so that the user of the data is informed as to what types of information are expected here. More information on these process parameters can be found in section 6.4 Process.

Recommendation 3 The name of each OM_Observation.parameter should correspond to a processParameter provided in the Process.

6.6.1 Specialised Observations

The ‘Specialised Observations’ package defines seven specialisations of OM_Observation (Figure 10).

All the specialised Observation types essentially add ‘constraints’ to the underlying O&M model which characterise the result of the observation and the sampling regime used¹. For example, a PointTimeSeriesObservation is a timeseries at a single point in space (e.g. at a fixed station), so the ‘Spatial Sampling Feature’ in 19156 must be a spatial sampling point, and the ‘phenomenonTime’ must be a time period i.e. the observation must be taken over a period of time. The type of the result must be a set of time, value pairs.

In actual fact, the specialised Observation types do not specialise OM_Observation directly but specialise the informative O&M class SpecialisedCoverageObservation, which in turn specialises DiscreteCoverageObservation. These two classes between them ensure that the result of the

¹ This pattern was modelled on the approach taken in Climate Science Modelling Language version 3 (OGC Pending Docs 11_021) which extends ISO 19156.
observation is a coverage, and the feature of interest is a ‘Spatial Sampling Feature’ e.g. a point, an area, a line.

![Specialised Observation Types Diagram](image-url)

Figure 10: Specialised Observation Types
6.6.1.1. **Point Observations**

6.6.1.1.1. **PointObservation**

The PointObservation (Figure 11) represents a single value measurement at a single point in time e.g. a manual one-off measurement of sea surface temperature.

![Class diagram for PointObservation](image-url)
### O&M Attribute/association | Constraint | Example: Single measurement of Air Temperature
--- | --- | ---
**processUsed** | Process (Section 5.2.1.1.10) | Process instance links to information about the responsible party, documented process etc.
**featureOfInterest** | SF_SamplingPoint | A *SF_SamplingPoint* at the geographic location of the measurement
**phenomenonTime** | TM_TimeInstant | A time instant (in ISO 8601 including time zone) e.g. 2012-01-30T10:30:00.00Z
**observedProperty** | ObservableProperty | The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
**result** | CV_DiscretePointCoverage | The result should be a single valued coverage recording an estimate of the observed property e.g. 22.2 (Celsius)
**resultTime** | TM_TimeInstant | The time the result was made available (e.g. published)

#### 6.6.1.1.2. *PointTimeSeriesObservation*

The *PointTimeSeriesObservation* (Figure 12) represents a series of measurements at the same point – a classic timeseries – e.g. regular measurements from a fixed station.

| O&M Attribute/association | Constraint | Example: Repeated measurements of Sea Surface Temperature at the same location.
--- | --- | ---
**processUsed** | Process | Process instance links to information about the responsible party, documented process etc.
**featureOfInterest** | SF_SamplingPoint | A *SF_SamplingPoint* at the geographic location of the measurement. It must be the same location for the entire time series. Note that in the case of fixed monitoring stations, the *SF_SamplingPoint* could be specialised in an extension schema to be a station feature type (or similar) to provide further information about the fixed station (e.g. a name). Although this is not required.
**phenomenonTime** | TM_TimePeriod | A time period (in ISO 8601) representing the start and end date/times of the time series.
**observedProperty** | ObservableProperty | The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
**result** | TimeSeries | The result should be a set of time,value pairs encoded according to the Generic Conceptual Model.
**resultTime** | TM_TimeInstant | The time the result was made available (e.g. published)
Figure 12: PointTimeSeriesObservation

6.6.1.1.3. MultiPointObservation

The MultiPointObservation (Figure 13) is a very specific type of Point-based observation. It is intended for the case where identical measurements are made at a set of discrete points at the same time. For example a sensor network reporting temperature at 10am. The points themselves are not on a grid but may be distributed in any manner – for example unevenly spaced around a coastline. In this case the result of the observation is a GML MultiPointCoverage, which consists of a set of points (the domain) and a set of values (the rangeSet). (see GML 3.3.3).
class MultiPointObservation

MultiPointObservation - Set of point measurements at same time instant, each point has location, value

- «FeatureType»
  - observation::OM_Observation

- «FeatureType»
  - coverageObservation::OM_DiscreteCoverageObservation

- «FeatureType»
  - SamplingCoverageObservation::SamplingCoverageObservation

  featureOfInterest.shape shall be consistent with spatial components of result domain

  observedProperty shall be consistent with result rangeType

  phenomenonTime shall be consistent with temporal component of result domain

- «featureType»
  - MultiPointObservation

  featureOfInterest shall be a SF_SamplingSurface
  /* featureOfInterest must be a SF_SamplingSurface */
  inv: self.featureOfInterest->forAll(oclIsKindOf(SF_SamplingSurface))

  result must be a MultiPointCoverage
  /* result must be a MultiPointCoverage */
  inv: self.result.oclIsKindOf(MultiPointCoverage)

  phenomenonTime shall be a TM_TimeInstant
  /* phenomenonTime must be a TM_TimeInstant */
  inv: self.phenomenonTime.oclIsKindOf(TM_TimeInstant)

---

Figure 13: MultiPointObservation

<table>
<thead>
<tr>
<th>O&amp;M Attribute/association</th>
<th>Constraint</th>
<th>Example: Repeated measurements of Sea Surface Temperature at the multiple locations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>processUsed</td>
<td>Process</td>
<td>Process instance links to information about the responsible party, documented process etc.</td>
</tr>
<tr>
<td>featureOfInterest</td>
<td>SF_SamplingSurface or SF_SamplingSolid</td>
<td>A SF_SamplingSurface with a geometry that defines the total extent of the MultiPointObservation (i.e. a bounding box or polygon that includes all the measurement locations).</td>
</tr>
<tr>
<td>phenomenonTime</td>
<td>TM_TimeInstant</td>
<td>A time instant (in ISO 8601) when the observations were taken (all measurements must be taken at the</td>
</tr>
</tbody>
</table>
same time instant).

<table>
<thead>
<tr>
<th>observedProperty</th>
<th>ObservableProperty</th>
<th>The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>MultiPointCoverage</td>
<td>The result should be a GML MultiPointCoverage. For large result sets an out-of-band result (e.g. in binary) may be provided.</td>
</tr>
<tr>
<td>resultTime</td>
<td>TM_TimeInstant</td>
<td>The time the result was made available (e.g. published).</td>
</tr>
</tbody>
</table>

### 6.6.1.2. Trajectory Observations

#### 6.6.1.2.1. ProfileObservation

A ProfileObservation (Figure 14) represents a set of points along a vertical axis with a measurement value at each point on the profile. The measurements are all nominally made at the same time for the entire profile. The profile is encoded as a one-dimensional grid coverage, again using the GML coverage models.

<table>
<thead>
<tr>
<th>O&amp;M Attribute/association</th>
<th>Constraint</th>
<th>Example: Salinity depth profile.</th>
</tr>
</thead>
<tbody>
<tr>
<td>processUsed</td>
<td>Process</td>
<td>Process instance links to information about the responsible party, documented process etc.</td>
</tr>
<tr>
<td>featureOfInterest</td>
<td>SF_SamplingCurve</td>
<td>A SF_SamplingCurve with a geometry that defines the geometry of the profile.</td>
</tr>
<tr>
<td>phenomenonTime</td>
<td>TM_TimeInstant</td>
<td>A time instant (in ISO 8601) when the observations were taken (all measurements must be taken at the same time instant).</td>
</tr>
<tr>
<td>observedProperty</td>
<td>ObservableProperty</td>
<td>The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).</td>
</tr>
<tr>
<td>result</td>
<td>RectifiedGridCoverage or ReferenceableGridCoverage</td>
<td>The result should be a GML RectifiedGridCoverage or ReferenceableGridCoverage with a single spatial dimension (which should be vertical). For large result sets an out-of-band result (e.g. in binary) may be provided.</td>
</tr>
<tr>
<td>resultTime</td>
<td>TM_TimeInstant</td>
<td>The time the result was made available (e.g. published)</td>
</tr>
</tbody>
</table>
Profile Observation - a series of measurements along a vertical line at the same nominal time

- featureOfInterest.shape shall be consistent with spatial components of result.domain
- observedProperty shall be consistent with result.rangeType
- phenomenonTime shall be consistent with temporal component of result.domain
- Profile is grid with one vertical axis
- spatial domain of the result shall contain one axis and that shall be vertical
- result must be a ReferenceableGridCoverage or RectifiedGridCoverage
  - result must be a ReferenceableGridCoverage or a RectifiedGridCoverage
  - featureOfInterest must be a SF_SamplingCurve
    - featureOfInterest must be a SF_SamplingCurve
      - featureOfInterest must be a SF_SamplingCurve

Figure 14: Profile Observation
6.6.1.2.2. TrajectoryObservation

A TrajectoryObservation (Figure 15) represents a series of measurements along a trajectory. For example along a ship’s track. Each measurement is made at a separate point along the trajectory and at a separate time. The result is therefore a set of time, location, value triples.

![Diagram of TrajectoryObservation](image-url)

**Figure 15: TrajectoryObservation**
### 6.6.1.3. Gridded Observations

#### 6.6.1.3.1. GridObservation

A GridObservation (Figure 16) is a single grid of data – e.g. measurements taken by a satellite processed to be on a rectified geo-referenced grid (e.g. Level 3 processed data), or output from a numerical model. The GridObservation is taken at a single snapshot in time. e.g. 10am, 30 January 2012.

<table>
<thead>
<tr>
<th>O&amp;M Attribute/association</th>
<th>Constraint</th>
<th>Example: Grid of Ocean Colour.</th>
</tr>
</thead>
<tbody>
<tr>
<td>processUsed</td>
<td>Process</td>
<td>Process instance links to information about the responsible party, documented process etc.</td>
</tr>
<tr>
<td>featureOfInterest</td>
<td>SF_SamplingSurface or SF_SamplingSolid (if there is a vertical dimension to the grid)</td>
<td>A SF_SamplingSurface that defines the extent of the Grid of data.</td>
</tr>
<tr>
<td>phenomenonTime</td>
<td>TM_Timelinst</td>
<td>A time instant (in ISO 8601 including time zone) e.g. 2012-01-30T10:30:00.00Z which the Grid represents.</td>
</tr>
<tr>
<td>observedProperty</td>
<td>ObservableProperty</td>
<td>The observed property should link to a vocabulary defining Ocean Colour, and should also indicate the units used in the result (e.g the index type).</td>
</tr>
<tr>
<td>result</td>
<td>RectifiedGridCoverage or ReferenceableGridCoverage</td>
<td>The result should be a GML RectifiedGridCoverage or GML ReferenceableGridCoverage containing the grid points (as the domain of the coverage) and the observed ocean colour values (as the rangeSet of the coverage. For large grids an out-of-band result (e.g. in binary) may be provided.</td>
</tr>
</tbody>
</table>
resultTime | TM_Timelinstant | The time the result was made available (e.g. published)

Figure 16: GridObservation

6.6.1.3.2. GridSeriesObservation

A GridSeriesObservation (Figure 17) is similar to a GridObservation except it contains a series of grids for multiple, successive timesteps (e.g. a simulation/model run)
class GridSeriesObservation

GridSeriesObservation

«FeatureType»
observation::OM_Observation

«FeatureType»
coverageObservation::OM_DiscreteCoverageObservation

«FeatureType»
SamplingCoverageObservation::SamplingCoverageObservation

featureOfInterest.shape shall be consistent with spatial components of result.domain

phenomenonTime shall be consistent with temporal component of result.domain

observedProperty shall be consistent with result.rangeType

phenomenonTime must be a TM_TimePeriod
/* phenomenonTime must be a TM_TimePeriod */
inv: self.phenomenonTime.oclIsKindOf(TM_TimePeriod)

result must be a RectifiedGridCoverage or ReferenceableGridCoverage
/* result must be a RectifiedGridCoverage or a ReferenceableGridCoverage */
inv: self.result.oclIsKindOf(RectifiedGridCoverage) OR self.result.oclIsKindOf(ReferenceableGridCoverage)

One of the axes of the domain must be a temporal axis

featureOfInterest must be a SF_SamplingSolid /* featureOfInterest must be a SF_SamplingSolid */
inv: self.featureOfInterest->forAll(oclIsKindOf(SF_SamplingSolid))

phenomenonTime must be a TM_Period /* phenomenonTime must be a TM_Period */
inv: self.phenomenonTime.oclIsKindOf(TM_Period)

result must be a RectifiedGridCoverage or ReferenceableGridCoverage
/* result must be a RectifiedGridCoverage or a ReferenceableGridCoverage */
inv: self.result.oclIsKindOf(RectifiedGridCoverage) OR self.result.oclIsKindOf(ReferenceableGridCoverage)

Figure 17: GridSeriesObservation

<table>
<thead>
<tr>
<th>O&amp;M Attribute/association</th>
<th>Constraint</th>
<th>Example: Grid of Ocean Colour.</th>
</tr>
</thead>
<tbody>
<tr>
<td>processUsed</td>
<td>Process</td>
<td>Process instance links to information about the responsible party, documented process etc.</td>
</tr>
<tr>
<td>featureOfInterest</td>
<td>SF_SamplingSurface or SF_SamplingSolid (if there is a vertical dimension to the grid)</td>
<td>A SF_SamplingSurface that defines the extent of the Grid of data.</td>
</tr>
<tr>
<td>phenomenonTime</td>
<td>TM_TimePeriod</td>
<td>A time period (in ISO 8601) representing</td>
</tr>
</tbody>
</table>
observedProperty | ObservableProperty | The observed property should link to a vocabulary defining Sea Surface Temperature, and should also indicate the units used in the result.

result | RectifiedGridCoverage or ReferenceableGridCoverage | The result should be a GML RectifiedGridCoverage or GML ReferenceableGridCoverage containing the grid points (as the spatio-temporal domain of the coverage) and the observed sea surface temperature values (as the rangeSet of the coverage). Note that one of the axes of the grid coverage domain must be a temporal axis as GridSeriesObservation is a type of time series. For detailed encoding of GML coverage types see GML 3.3.3. For large grids an out-of-band result (e.g. in binary) may be provided.

resultTime | TM_TimeInstant | The time the result was made available (e.g. published)

6.6.2 Feature catalogue

Feature catalogue metadata

<table>
<thead>
<tr>
<th>Application Schema</th>
<th>INSPIRE Application Schema Specialised Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version number</td>
<td>2.0rc3</td>
</tr>
</tbody>
</table>

Types defined in the feature catalogue

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Stereotypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridObservation</td>
<td>Gridded Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>GridSeriesObservation</td>
<td>Gridded Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>MultiPointObservation</td>
<td>Point Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>PointObservation</td>
<td>Point Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>PointObservationCollection</td>
<td>Point Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>PointTimeSeriesObservation</td>
<td>Point Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>ProfileObservation</td>
<td>Trajectory and Profile Observations</td>
<td>«featureType»</td>
</tr>
<tr>
<td>TimeLocationValueTriple</td>
<td>Trajectory and Profile Observations</td>
<td>«dataType»</td>
</tr>
<tr>
<td>TrajectoryObservation</td>
<td>Trajectory and Profile Observations</td>
<td>«featureType»</td>
</tr>
</tbody>
</table>

6.6.2.1. Spatial object types

6.6.2.1.1. GridObservation

GridObservation

<table>
<thead>
<tr>
<th>Name:</th>
<th>GridObservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtype of:</td>
<td>SamplingCoverageObservation</td>
</tr>
<tr>
<td>Definition:</td>
<td>Observation representing a gridded field at a single time instant.</td>
</tr>
<tr>
<td>Description:</td>
<td>A GridObservation is an observation of some phenomenon (or phenomena) over a gridded field. E.g. Output from a model, or rectified, georeferenced satellite data. The result of a GridObservation is a discrete coverage within a compound spatiotemporal CRS where the domain consists of a two- or three-dimensional grid of points, all having the same time instant temporal component.</td>
</tr>
</tbody>
</table>
GridObservation

- Stereotypes: «featureType»

**Constraint: featureOfInterest must be a SF_SamplingSolid or SF_SamplingSurface**

- Natural language: featureOfInterest must be a SF_SamplingSolid or SF_SamplingSurface
- OCL: inv: self.featureOfInterest->forall(oclIsKindOf(SF_SamplingSolid)) OR inv: self.featureOfInterest->forall(oclIsKindOf(SF_SamplingSurface))

**Constraint: phenomenonTime must be a TM_Instant**

- Natural language: phenomenonTime must be a TM_Instant
- OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Instant)

**Constraint: result must be a RectifiedGridCoverage or ReferenceableGridCoverage**

- Natural language: result must be a RectifiedGridCoverage or ReferenceableGridCoverage
- OCL: inv: self.result.oclIsKindOf(RectifiedGridCoverage) OR self.result.oclIsKindOf(ReferenceableGridCoverage)

6.6.2.1.2. GridSeriesObservation

**GridSeriesObservation**

- Name: GridSeriesObservation
- Subtype of: SamplingCoverageObservation
- Definition: Observation representing an evolving gridded field at a succession of time instants.
- Description: A GridSeriesObservation is a time series of gridded fields representing the same phenomenon (or phenomena) over a series of times. E.g. Ocean model output. The result of a GridSeriesObservation is a discrete coverage within a compound spatiotemporal CRS where the domain consists of a series of two- or three-dimensional grids of points, each at a successive time instant.
- Stereotypes: «featureType»

**Constraint: featureOfInterest must be a SF_SamplingSolid**

- Natural language: featureOfInterest must be a SF_SamplingSolid
- OCL: inv: self.featureOfInterest->forall(oclIsKindOf(SF_SamplingSolid))

**Constraint: One of the axes of the domain must be a temporal axis.**

- Natural language: 
- OCL:

**Constraint: phenomenonTime must be a TM_Period**

- Natural language: phenomenonTime must be a TM_Period
- OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Period)

**Constraint: result must be a RectifiedGridCoverage or ReferenceableGridCoverage**

- Natural language: result must be a RectifiedGridCoverage or a ReferenceableGridCoverage
- OCL: inv: self.result.oclIsKindOf(RectifiedGridCoverage) OR self.result.oclIsKindOf(ReferenceableGridCoverage)

6.6.2.1.3. PointObservation

**PointObservation**
**PointObservation**

**Name:** Point Observation  
**Subtype of:** SamplingCoverageObservation  
**Definition:** Observation that represents a measurement of a property at a single point in time and space.  
**Description:** The PointObservation represents a single measurement or estimation of a property at a single point in time and space. For example a single temperature measurement at a fixed weather station.  
**Stereotypes:** «featureType»

**Constraint: featureOfInterest must be a SF_SamplingPoint**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.featureOfInterest-&gt;forall(oclIsKindOf(SF_SamplingPoint))</th>
</tr>
</thead>
</table>

**Constraint: phenomenonTime must be a TM_Instant**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Instant)</th>
</tr>
</thead>
</table>

**Constraint: result must be a CV_DiscretePointCoverage**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.result.oclIsKindOf(CV_DiscretePointCoverage)</th>
</tr>
</thead>
</table>

**6.6.2.1.4. MultiPointObservation**

**MultiPointObservation**

**Name:** MultiPointObservation  
**Subtype of:** SamplingCoverageObservation  
**Definition:** Observation that represents a set of measurements all made at exactly the same time but at different locations  
**Description:** The MultiPointObservation is an Observation that represents a set of measurements all made at exactly the same time but at different locations, for example a distributed sensor network reporting the temperature at 10am. The result of this observation is a MultiPointCoverage.  
**Stereotypes:** «featureType»

**Constraint: featureOfInterest shall be a SF_SamplingSurface**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.featureOfInterest-&gt;forall(oclIsKindOf(SF_SamplingSurface))</th>
</tr>
</thead>
</table>

**Constraint: phenomenonTime shall be a TM_Instant**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Instant)</th>
</tr>
</thead>
</table>

**Constraint: result must be a MultiPointCoverage**

<table>
<thead>
<tr>
<th>Natural language</th>
<th>OCL: inv: self.result.oclIsKindOf(MultiPointCoverage)</th>
</tr>
</thead>
</table>

**6.6.2.1.5. PointTimeSeriesObservation**

**PointTimeSeriesObservation**

**Name:** PointTimeSeriesObservation  
**Subtype of:** SamplingCoverageObservation
PointTimeSeriesObservation

Definition: Observation that represents a time-series of point measurements of a property at a fixed location in space

Description: A PointTimeSeriesObservation is a time series of observations made at the same fixed spatial location e.g. Measurements made repeatedly by a fixed monitoring instrument.

Stereotypes: «featureType»

Constraint: featureOfInterest must be a SF_SamplingPoint
Natural language: featureOfInterest must be a SF_SamplingPoint
OCL: inv: self.featureOfInterest->forall(oclIsKindOf(SF_SamplingPoint))

Constraint: phenomenonTime must be a TM_Period
Natural language: phenomenonTime must be a TM_Period
OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Period)

Constraint: result must be a TimeSeries
Natural language: result must be a TimeSeries
OCL: inv: self.result.oclIsKindOf(TimeSeries)

6.6.2.1.6. PointObservationCollection

PointObservationCollection

Name: PointObservationCollection
Subtype of: ObservationSet
Definition: A collection of Point Observations.
Description: The PointObservationCollection is a collection of separate PointObservations. In the case where it is useful to group together a set of otherwise independent PointObservations the PointObservationCollection should be used to make this grouping. The grouping may be made on any basis e.g. it may be useful to group together PointObservations made by the same instrument or Environmental Facility, or in a particular measurement campaign. Each member of the PointObservationCollection must be a single PointObservation.

Stereotypes: «featureType»

Constraint: member shall be of type PointObservation
Natural language: each member shall be a PointObservation
OCL: inv: self.member.oclIsKindOf(PointObservation)

6.6.2.1.7. ProfileObservation

ProfileObservation

Name: ProfileObservation
Subtype of: SamplingCoverageObservation
Definition: Observation representing the measurement of a property along a vertical profile in space at a single time instant.
Description: A ProfileObservation is an Observation representing the measurement of a property along a vertical profile in space at a single time instant. For example a CTD profile measuring salinity at different depths in the ocean.

Stereotypes: «featureType»

Constraint: featureOfInterest must be a SF_SamplingCurve
Natural language: featureOfInterest must be a SF_SamplingCurve
ProfileObservation

OCL: inv: self.featureOfInterest->forAll(oclIsKindOf(SF_SamplingCurve))

Constraint: phenomenonTime must be a TM_Instant

Natural language: phenomenonTime must be a TM_Instant
OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Instant)

Constraint: result must be a ReferenceableGridCoverage or RectifiedGridCoverage

Natural language: result must be a ReferenceableGridCoverage or a RectifiedGridCoverage
OCL: inv: self.result.oclIsKindOf(ReferenceableGridCoverage) OR inv: self.result.oclIsKindOf(RectifiedGridCoverage)

Constraint: spatial domain of the result shall contain one axis and that shall be vertical

Natural language: OCL:

6.6.2.1.8. TrajectoryObservation

TrajectoryObservation

Name: TrajectoryObservation
Subtype of: SamplingCoverageObservation
Definition: Observation representing the measurement of a property along a meandering curve in time and space.
Description: A TrajectoryObservation is an Observation representing the measurement of a property along a meandering curve in time and space. For example a Pollutant concentration from a mobile air quality sensor.
Stereotypes: «featureType»

Constraint: featureOfInterest must be a SF_SamplingCurve

Natural language: featureOfInterest must be a SF_SamplingPoint
OCL: inv: self.featureOfInterest->forAll(oclIsKindOf(SF_SamplingPoint))

Constraint: phenomenonTime must be a TM_Period

Natural language: phenomenonTime must be a TM_Period
OCL: inv: self.phenomenonTime.oclIsKindOf(TM_Period)

Constraint: result must be a TimeSeries

Natural language: result must be a TimeSeries
OCL: inv: self.result.oclIsKindOf(TimeSeries)

Constraint: result.point must be TimeLocationValueTriple

Natural language: each point in the result must be a TimeLocationValueTriple
OCL: inv: self.result.point.oclIsKindOf(TimeLocationValueTriple)

6.6.2.2. Data types

6.6.2.2.1. TimeLocationValueTriple

TimeLocationValueTriple

Name: TimeLocationValue Triple
Subtype of: AnnotatedTimeValuePair
TimeLocationValueTriple

- **Definition:** A triple set of Time, location, value (measurement). For example, at a point along a trajectory.
- **Stereotypes:** «dataType»

**Attribute: location**

- **Name:** location
- **Value type:** GM_Position
- **Definition:** Geographic location where value is valid.
- **Multiplicity:** 1

### 6.6.2.3. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

#### 6.6.2.3.1. AnnotatedTimeValuePair

**AnnotatedTimeValuePair**

- **Package:** Timeseries

#### 6.6.2.3.2. GM_Position

**GM_Position**

- **Package:** Coordinate geometry
- **Reference:** Geographic information -- Spatial schema [ISO 19107:2003]

#### 6.6.2.3.3. ObservationSet

**ObservationSet**

- **Package:** Observation References
- **Reference:** Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE [DS-D2.9]
- **Definition:** Links a set of Observations
- **Description:** This class is used to link multiple related Observations together

#### 6.6.2.3.4. SamplingCoverageObservation

**SamplingCoverageObservation**

- **Package:** Sampling Coverage Observation
6.7 Referencing to and from Observations

The core of this document addresses INSPIRE themes which directly integrate O&M into their data specifications. However, there are also themes whose where observational data is not the main focus but where the final resulting data provided under INSPIRE is based on observational data. In such cases, it would be valuable to reference the source observations used in the generation of the results provided.

Such references can be provided to individual Observations using a simple association. An example for such an association (from the feature type AbstractMonitoringFeature, defined in the theme “Environmental Monitoring Facilities”) is shown in Figure 18. In addition, the featureType ObservationSet has been defined for the semantic grouping of Observations. Multiple observations being referred to as a set can first be grouped and then referenced by other themes referencing groups of observations.

![Diagram of class ReferencesToObservation](image)

Figure 18: Direct association to observations and observation sets

In other cases, Observations are provided but not directly linked from the Environmental Monitoring Facilities at which they were made. To establish the link to the Environmental Monitoring Facility, the parameter attribute of the Observation shall be used as specified in the following IR Requirement.

### IR Requirement
Annex I, Section 7.5
Requirements for Observations

Where the OM_Observation type or any sub-type thereof is used to make data available, the following requirements shall apply:

(…)

(2) Where reference is made to an EnvironmentalMonitoringFacility from an OM_Observation, a parameter attribute shall be provided, whose name attribute is "relatedMonitoringFeature" and whose value attribute is of type AbstractMonitoringFeature.
EXAMPLE  The object diagram in Figure 19 shows an example for how to use the parameter attribute for referring to a related monitoring feature.

```
MyObservation123 : OM_Observation

parameter=
    name = "relatedMonitoringFeature"
    value = "EF_123"
...
...
EF_123 : EMF
```

Figure 19: Linkage with parameter attribute

6.7.1 Feature catalogue

Feature catalogue metadata

Application Schema | INSPIRE Application Schema Observation References
Version number     | 2.0rc3

Types defined in the feature catalogue

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Stereotypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObservationSet</td>
<td>Observation References</td>
<td>«featureType»</td>
</tr>
</tbody>
</table>

6.7.1.1. Spatial object types

6.7.1.1.1. ObservationSet

ObservationSet

<table>
<thead>
<tr>
<th>Name:</th>
<th>ObservationSet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Links a set of Observations</td>
</tr>
<tr>
<td>Description:</td>
<td>This class is used to link multiple related Observations together</td>
</tr>
<tr>
<td>Stereotypes:</td>
<td>«featureType»</td>
</tr>
</tbody>
</table>

Attribute: extent

<table>
<thead>
<tr>
<th>Name:</th>
<th>extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value type:</td>
<td>EX_Extent</td>
</tr>
<tr>
<td>Definition:</td>
<td>Information about the spatial and temporal extent.</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>1</td>
</tr>
</tbody>
</table>

Attribute: inspireId

<table>
<thead>
<tr>
<th>Name:</th>
<th>inspireId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value type:</td>
<td>Identifier</td>
</tr>
<tr>
<td>Definition:</td>
<td>External object identifier of the spatial object.</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>1</td>
</tr>
</tbody>
</table>

Association role: member

<table>
<thead>
<tr>
<th>Name:</th>
<th>member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value type:</td>
<td>OM_Observation</td>
</tr>
<tr>
<td>Definition:</td>
<td>One member of the ObservationSet</td>
</tr>
<tr>
<td>Multiplicity:</td>
<td>1..*</td>
</tr>
</tbody>
</table>

6.7.1.2. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.
6.7.1.2.1.  **EX_Extent**

**EX_Extent**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Extent information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference:</td>
<td>Geographic information -- Metadata [ISO 19115:2003/Cor 1:2006]</td>
</tr>
</tbody>
</table>

6.7.1.2.2.  **Identifier**

**Identifier**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Base Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference:</td>
<td>INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5]</td>
</tr>
<tr>
<td>Definition:</td>
<td>External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.</td>
</tr>
<tr>
<td>Description:</td>
<td>NOTE1 External object identifiers are distinct from thematic object identifiers.</td>
</tr>
</tbody>
</table>

NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same object.

NOTE 3 The unique identifier will not change during the life-time of a spatial object.

6.8  **Facility**

While there is no formal facility or station concept within the O&M standard, there is often a requirement to provide information on specialized locations used for the provision of multiple observations. While the O&M standard suggests the modelling of stations, which could be seen as facilities, as a form of SamplingPoint, this causes difficulties when one wishes to include remote sensing within the facility concept. It also doesn’t provide support for mobile facilities, and thus cannot be used within the INSPIRE context.

Within INSPIRE, Environmental Monitoring Facilities are their own theme across thematic domains. At the same time, other INSPIRE Themes provide either primary data from Environmental Monitoring Facilities, or processes results based on this primary data. Thus, a facility concept is being defined to cover the requirements stemming from all these themes.

This wide range of facility concepts requires a model that allows for reuse of the facility concept in various concepts, supporting various requirements. These requirements range from very simple information on where the facility is located and what media is being assayed over more detailed information on exactly what phenomena are being monitored at this facility at which locations using which localities to cases where the full set of observational data obtained from the facility must be provided.

INSPIRE Environmental Monitoring Facilities can also be grouped together into Networks. As derived coverage results are often calculated from primary measurements stemming from an entire network, this type of observation should be linked to the Network class.

6.8.1  **Observing Capability of a Facility**

Within TWG EF, the facility type has been modeled with a cascadable relationship allowing for multiple facility parts to be nested under a higher level facility. This has been modeled in this manner as a station can have various parts or platforms, in turn containing further parts, which at the end contain an array of sensors or other measurement devices. At whatever level of this cascade, it must be possible to attach measurement data as well as a clear description of the measurement regime, which includes:

- Result Location (Where)
- Observed Property (What)
- Method (How)
A difficulty arises because there are cases where the measurement regime must be provided while the actual measurements will not be provided. For this purpose, we have defined the ObservingCapability Class, that serves to link the measurement regime to the facility class.

![Diagram of Observing Capability Class]

Figure 20: Observing Capability Class

ObservingCapability Class has been defined with the following attributes:

- **observingTime**: TM_Object – the time that this measurement regime is operational. This may be only the starting date for facilities that are still in operation, or a time interval for facilities that only provided data for a specific interval in time.

- **processType**: ProcessTypeValue – as we allow for encoding of the OM_Process either as an Process or via SensorML, this attribute shows the user what data type is to be expected under the process relation. Should other forms of process encoding be used, this should be added to the codelist.

- **resultNature**: ResultNatureValue – describes the nature of the result, the following options are available:
  - primary
  - processed
  - simulated

In addition to these attributes, the ObservingCapability Class has the following associations:

- **ProcessUsed**
- **Phenomenon**
- **Domain**

The definitions of these relations are the same as the corresponding ones from the OM_Observation.
The Facility and Network types defined for the Environmental Monitoring Facilities Theme provide the constraints to assure that observations may only be linked to a facility that has an ObservingCapability Class referencing the same Procedure, ObservedProperty and Feature-of-Interest as those referenced in the Observation itself.

6.8.2 Further Observation Related Facility Attributes

Pertaining to the question: ‘what is the location of the Facility?’ - ‘what is the location of the Observation?’ - ‘what is the location of the ultimate FoI?’, this can be decompose into the following three options for the result acquisition source:

- **in situ**: the the sampling feature is co-located with the ultimate FOI (i.e. the sampledFeature). the measurement process is performed at this location
- **ex situ**: the sampling feature is a specimen taken from the ultimate FOI (i.e. the sampledFeature), the measurement process is performed at a different location
- **remote**: the sampling feature is often the ultimate FOI (i.e. the sampledFeature), the measurement process is performed at a different location (but no sample taken). This also be used for simulations.

A further question that comes up is the nature of the result. While classical observational data tends to be primary measurement or observational data, O&M is also well suited to the provision of a highly processed or aggregated nature; this leads to a differentiation of the result nature into the following three categories:

- **Primary**: The result provided with the observation is the direct result of an estimate of a property on the feature-of-interest. No further processing has been performed. Processing may have taken place, but only in the sense of the measurement methodology itself, i.e. converting the millivolt returned from the sensor to the concentration of a substance.
- **Processed**: The result provided, while usually based on primary measurements, has been substantially processed. This processing can be of diverse natures, in some situations complex aggregates are provided, in other situations, the existing values are interpolated to a continuum.
- **Simulated**: The result provided, while usually based on primary measurements, is based on an interpretation model, and provides a simulation of past or future states of the media being analyzed. In this case, the existing values are usually extrapolated into the past or future.

The constellation of objects related to an observation, and the ensuing possible design patterns, can also be influenced by the fact that the facility performing the measurement is fixed or mobile. In a fixed facility, there is a strong coupling between the facility and the FoI; with a moving facility, while the FoI is often co-located with the current location of the facility, the locational information cannot be directly linked to the location of the facility. Thus, differentiation into fixed and mobile facilities must be made. It must also be possible to mark Facilities as fixed or mobile, as mobile facilities will not have a fixed location while fixed facilities can be expected to contain this.

7 Metadata

Various elements describing the observation would be valuable for discovery purposes, and thus should be added to the Metadata requirements. This is a highly domain oriented question, closely related to architecture issues (see OGC Water Information Services Concept Development Study Report; CSW ebRIM to harvest SOS service) Concepts that could be included in the metadata are:

- overarching EF info (Facility, Network, MonitoringProgram),
- Observed property,
- Specific measurement method,
- dateTime info,
- ...
Which concepts to finally include as well as where they should be mapped to within MD_MetaData shall be determined after the Version 2 Testing phase.

8 Provision of O&M encoded data

<table>
<thead>
<tr>
<th>IR Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annex I, Section 7.5</td>
</tr>
<tr>
<td>Requirements for Observations</td>
</tr>
</tbody>
</table>

Where the OM_Observation type or any sub-type thereof is used to make data available, the following requirements shall apply:

(3) For all encodings that are used for all or part of an OM_Observation result, a public Application Programming Interface (API) shall be available to read the encoded file. This API shall be capable of exposing the information needed to realise INSPIRE spatial objects.

8.1 Services for the Provision of O&M Encoded Data

While it is possible to serve the various O&M related classes via a WFS or WCS, this is not the most comfortable method of accessing this data. We recommend the inclusion of the Sensor Observation Service (SOS 2.0) into the Network Services to be used for download services.

8.2 Registers/Registries

The requirement for various types of Registers/Registries/Codelists is very strong in the O&M area. These registers will usually need to be extendible. An additional challenge is the requirement for various types of structured registers, i.e. hierarchies and cross links such as found in thesauri or taxonomies. While we are aware of the fact that Registers/Registries/Codelists will also be required within other areas of INSPIRE, we wish to stress the fact that this is a very strong requirement for the clear provision of re-usable observational data.
Annex A
(informative)

Encoding

A.1 SWE Common Results

The following section shows how a single result can be provided via the use of the result types provided by OGC SWE Common. In the first example, we provide a measurement as the result (concentration of ozone in µg/m³):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns:om="http://www.opengis.net/om/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:swe="http://www.opengis.net/swe/2.0"
gml:id="oc_0" xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd http://www.opengis.net/om/2.0
http://schemas.opengis.net/sweCommon/2.0/swe.xsd http://www.opengis.net/sampling/2.0
http://schemas.opengis.net/sampling/2.0/samplingFeature.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
      <gml:lowerCorner>46.679722 9.653712777777779</gml:lowerCorner>
      <gml:upperCorner>48.8786111 17.0713889</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <om:OM_Observation gml:id="Obs1">
      <om:phenomenonTime>
        <gml:Timelinst gml:id="Time1">
          <gml:timePosition>2008-09-01T02:00:00+0200</gml:timePosition>
        </gml:Timelinst>
      </om:phenomenonTime>
      <om:resultTime>
        <gml:Timelinst gml:id="Time2">
          <gml:timePosition>2008-09-01T05:00:00+0200</gml:timePosition>
        </gml:Timelinst>
      </om:resultTime>
      <om:observedProperty xlink:href="urn:ogc:def:phenomenon:a1t:O3"/>
      <om:featureOfInterest>
        <gml:FeatureCollection gml:id="FoI1">
          <gml:featureMember xlink:href="urn:ogc:object:feature:a1t:09:AT0098A"/>
        </gml:FeatureCollection>
      </om:featureOfInterest>
      <om:result>
        <swe:Quantity>
          <swe:uom code="ug/m2"/>
          <swe:value>28.614536</swe:value>
        </swe:Quantity>
      </om:result>
    </om:OM_Observation>
  </gml:featureMember>
</gml:FeatureCollection>
```
In the second example, we again provide a single result; however, in this example the result is not coming from a measurement, but is the result of an observation determining the species of an individual tree. Thus, we use a Category type for this type of categorization, and additionally provide the codespace this entry was taken from, as well as a human readable description in addition to the GUID uniquely identifying this species within the codespace:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns:om="http://www.opengis.net/om/2.0"
gml:id="oc_0" xsi:schemaLocation="http://www.opengis.net/gml/3.2 http://schemas.opengis.net/gml/3.2.1/gml.xsd http://www.opengis.net/om/2.0 http://schemas.opengis.net/om/2.0/observation.xsd http://www.opengis.net/swe/2.0 http://schemas.opengis.net/sweCommon/2.0/swe.xsd http://www.opengis.net/sampling/2.0 http://schemas.opengis.net/sampling/2.0/samplingFeature.xsd">
<gml:boundedBy>
  <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
    <gml:lowerCorner>46.679722 9.653712777777779</gml:lowerCorner>
    <gml:upperCorner>48.8786111 17.0713889</gml:upperCorner>
  </gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
  <om:OM_Observation gml:id="Obs1">
    <om:phenomenonTime>
      <om:TimeInstant gml:id="Time1">
        <gml:timePosition>2008-09-01T02:00:00+0200</gml:timePosition>
      </om:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime>
      <om:TimeInstant gml:id="Time2">
        <gml:timePosition>2008-09-01T05:00:00+0200</gml:timePosition>
      </om:TimeInstant>
    </om:resultTime>
    <om:observedProperty xlink:href="urn:ogc:def:phenomenon:species:identification"/>
    <om:featureOfInterest>
      <gml:FeatureCollection gml:id="FoI1">
      </gml:FeatureCollection>
    </om:featureOfInterest>
    <om:result>
      <swe:Category>
        <swe:description>Fagus sylvatica L.</swe:description>
        <swe:value>68C1AC04-391B-49DF-990A-3DD6A75D05B6</swe:value>
      </swe:Category>
    </om:result>
  </om:OM_Observation>
</gml:featureMember>
</gml:FeatureCollection>
```
The following section shows how a time series can be provided via the use of the result types provided by OGC SWE Common. In this example the observation pertains to hourly ozone measurements; the time series provided has been truncated to 3 values for brevity:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection
xmlns:om="http://www.opengis.net/om/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:swe="http://www.opengis.net/swe/2.0"
  <gml:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
      <gml:lowerCorner>46.679722 9.653712777777779</gml:lowerCorner>
      <gml:upperCorner>48.8786111 17.0713889</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <om:OM_Observation gml:id="Obs1">
      <om:phenomenonTime>
        <gml:TimePeriod gml:id="Time1" xsi:type="gml:TimePeriodType">
          <gml:beginPosition>2008-09-01T02:00:00+0200</gml:beginPosition>
          <gml:endPosition>2008-09-01T05:00:00+0200</gml:endPosition>
        </gml:TimePeriod>
      </om:phenomenonTime>
      <om:resultTime>
        <gml:TimeInstant gml:id="Time2">
          <gml:timePosition>2008-09-01T05:00:00+0200</gml:timePosition>
        </gml:TimeInstant>
      </om:resultTime>
      <om:observedProperty xlink:href="urn:ogc:def:phenomenon:ait:O3"/>
      <om:featureOfInterest>
        <gml:FeatureCollection gml:id="FoI1">
        </gml:FeatureCollection>
      </om:featureOfInterest>
      <om:result>
        <swe:DataArray>
          <swe:elementCount>
            <swe:Count>
              <swe:value>3</swe:value>
            </swe:Count>
          </swe:elementCount>
          <swe:elementType name="Components">
            <swe:DataRecord>
              <swe:field name="Time">
                <swe:Time definition="http://www.opengis.net/def/property/OGC/0/SamplingTime">
                </swe:Time>
              </swe:field>
              <swe:field name="feature">
                <swe:Text definition="urn:ogc:data:feature"/>
              </swe:field>
              <swe:field name="O3">
                <swe:Quantity definition="urn:ogc:def:phenomenon:ait:O3">
                  <swe:uom code="ppb"/>
                </swe:Quantity>
              </swe:field>
            </swe:DataRecord>
          </swe:elementType>
        </swe:DataArray>
      </om:result>
    </om:OM_Observation>
  </gml:featureMember>
</gml:FeatureCollection>
```
A.2 GML Coverage Results

The following section shows how a GML Coverage can be provided as an Observation result:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Gravity Processing Campaign encoded as multipoint coverage -->
<gmlcov:MultiPointCoverage gml:id="mpc-1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0" xmlns:swe="http://www.opengis.net/swe/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/gmlcov/1.0 http://schemas.opengis.net/gmlcov/1.0/coverage.xsd">
  <gml:multiPointDomain>
    <gml:Point gml:id="mp-1" srsDimension="2" srsName="EPSG:4326">
      <gml:pointMember>
        <gml:Point gml:id="stn-001">
          <gml:pos>654543 76674</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gml:id="stn-002">
          <gml:pos>654553 76634</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gml:id="stn-003">
          <gml:pos>654573 76654</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gml:id="stn-004">
          <gml:pos>654593 76624</gml:pos>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gml:id="stn-005">
          <gml:pos>654533 76614</gml:pos>
        </gml:Point>
      </gml:pointMember>
    </gml:Point>
  </gml:multiPointDomain>
  <gml:rangeSet>
    <gml:DataBlock>
      ... (additional data blocks) ...
    </gml:DataBlock>
  </gml:rangeSet>
</gmlcov:MultiPointCoverage>
```
<gml:rangeParameters/>
<gml:tupleList cs="n" ts="">
<stn-001 980000 980000 980000 0 0
<stn-002 980000 980000 980000 0 0
<stn-003 980000 980000 980000 0 0
<stn-004 980000 980000 980000 0 0
<stn-005 980000 980000 980000 0 0</gml:tupleList>
</gml:DataBlock>
</gml:rangeSet>
<gmlcov:rangeType>
<swe:DataRecord id="drec-1">
<swe:field name="identifier">
<swe:Text>
<swe:identifier/>
</swe:Text>
</swe:field>
<swe:field name="observedGravity">
<swe:uom code="microGal"/>
</swe:Quantity>
</swe:field>
<swe:field name="gravityFreeAirAnomaly">
<swe:uom code="microGal"/>
</swe:Quantity>
</swe:field>
<swe:field name="gravityBouguerAirAnomaly">
<swe:uom code="microGal"/>
</swe:Quantity>
</swe:field>
<swe:field name="innerTopoCorrection">
<swe:uom code="microGal"/>
</swe:Quantity>
</swe:field>
<swe:field name="totalTopoCorrection">
<swe:uom code="microGal"/>
</swe:Quantity>
</swe:field>
</swe:DataRecord>
</gmlcov:rangeType>
</gmlcov:MultiPointCoverage>
A.3 SensorML Process

The following section shows how an INSPIRE Process can be encoded using OGC SensorML 1.0.1 provided by OGC SWE Common. The mapping from the INSPIRE Process to SensorML is shown in section “6.4.2 Process Encoding with SensorML”:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sml:SensorML version="1.0.1" xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sensorML/1.0.1 http://schemas.opengis.net/sensorML/1.0.1/system.xsd">
  <sml:member>
    <sml:Component>
      <sml:Identification>
        <sml:IdentifierList>
          <sml:identifier name="inspireID">
            <sml:Term>
              <sml:codeSpace xlink:href="namespace"/>
              <sml:value>identifier</sml:value>
            </sml:Term>
          </sml:identifier>
        </sml:IdentifierList>
        <sml:Identification>
          <sml:IdentifierList>
            <sml:identifier name="procedureName">
              <sml:Term>
                <sml:value>procedureName</sml:value>
              </sml:Term>
            </sml:identifier>
          </sml:IdentifierList>
          <sml:Classification>
            <sml:ClassifierList>
              <sml:classifier name="type">
                <sml:Term>
                  <sml:value>procedureType, i.e. "raingauge"</sml:value>
                </sml:Term>
              </sml:classifier>
            </sml:ClassifierList>
            <sml:Classification>
              <sml:ClassifierList>
                <sml:classifier name="description">
                  <sml:Term>
                    <sml:value>the water hardness of the river being sampled from</sml:value>
                  </sml:Term>
                </sml:classifier>
              </sml:ClassifierList>
              <sml:Classification>
                <sml:ClassifierList>
                  <sml:classifier name="name">
                    <sml:Term>
                      <sml:value>WaterHardness</sml:value>
                    </sml:Term>
                  </sml:classifier>
                </sml:ClassifierList>
              </sml:Classification>
            </sml:Classification>
            <sml:Contact>
              <sml:ResponsibleParty>
                <sml:individualName>Alexandre Robin</sml:individualName>
              </sml:ResponsibleParty>
            </sml:Contact>
          </sml:Identification>
        </sml:Identification>
      </sml:Component>
    </sml:member>
  </sml:SensorML>
```
<table>
<thead>
<tr>
<th>INSPIRE</th>
<th></th>
<th></th>
<th>Reference: D2.9 v2.0rc3</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-TWG-OM</td>
<td>Guidelines for the use of Observations &amp; Measurements</td>
<td>2013-02-04</td>
<td>Page 79</td>
</tr>
</tbody>
</table>

- **University of Alabama in Huntsville**
- **Research Scientist**
- **Phone:** (256)817978
- **Electronic Mail Address:** robin@nsstc.uah.edu

**Document**
- **Description:** Description of Process
- **Date:** 2010-12-02
- **Online Resource:** [http://www.umweltbundesamt.at/](http://www.umweltbundesamt.at/)
Annex B
(informative)

Discussion Paper on Out-of-Band Results

DISCUSSION PAPER ON

Harmonising the delivery of gridded data as OM_Observation result in the INSPIRE Data Specifications based on the Observations & Measurements data model

Version 1.3
21st February 2012

Ilkka Rinne / Spatineo Inc.
on behalf of the Finnish Meteorological Institute
ilkka.rinne@spatineo.com

With review and contributions by
Dominic Lowe / British Atmospheric Data Centre, UK
Science & Technology Facilities Council
dominic.lowe@stfc.ac.uk
B.1 Introduction

INSPIRE Data Specification themes like Atmospheric Conditions and Meteorological Geographical Features as well as in Oceanographic Geographical Features have come to a conclusion that for large grid coverage type INSPIRE data sets of those themes, it must be possible to link from GML encoded OM_Observation instances to results of those Observations encoded in external resources encoded using binary data formats.

The version 2.9 of the INSPIRE AC-MF Data Specification summarises the out-of-band result delivery needs as follows:

“Due to the very large volumes of items in many AC-MF data sets, the encoding of the grid coverage result data of these data sets is typically stored in highly optimised binary file formats. Encoding of such data in textual form, like GML, would in most case result in file sizes impractical to produce, transfer over the network or consume by the users.”

This kind of linking either the entire value of the GML encoded OM_Observations’ result properties, or parts of them, to external files providing the actual values, is called here “out-of-band result delivery”. On the other hand providing the entire OM_Observation, including the value of the result property, using the same encoding, is called here “in-band result delivery”.

The purpose of this discussion paper is to provide background information on the practical issues for delivering spatial objects based on the data model of the Observations & Measurements (ISO 19156:2011) standard, and listing some in-band and out-of-band techniques for delivering O&M Observations with the grid coverage type results using INSPIRE Download Service. This paper is intended to provide additional information to the delivery and encoding sections of the INSPIRE technical guidance document “D2.9 Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development”.

B.2 Binding existing grid coverage data sets to OM_Observations

As defined in the Implementing Rules of INSPIRE Download Services, there are two possibilities for implementing an INSPIRE Download Service: offering pre-defined data sets for download or providing a “direct access” download service with query capabilities. In many cases the data sets in the INSPIRE Themes using the O&M based data models are collections of feature instances of class OM_Observation or classes inherited from OM_Observation.

Collections of the OM_Observations served using an INSPIRE Download Service can be either pre-defined using semantic grouping (like all measurement events with all measured properties from a single meteorological observation station within one day), or be created ad-hoc as a result of the given selection criteria for the Get Spatial Objects function of a direct-access Download Service. It’s recommended that the encoding and packaging of the OM_Observation instances for both direct access and non-direct access INSPIRE Download Services should as similar as possible. For the end-user’s perspective accessing an non-direct access Download Service should be like accessing a direct access Download Service with a pre-defined query.

The central idea of the INSPIRE Directive must be kept in mind: the INSPIRE Web Services exist for easy and harmonised access for geospatial information across both national and thematic borders within the EU. Thus the data sets must be harmonised as much as possible within the Data Specifications for each INSPIRE theme. Further considerable financial savings can be made if the data delivery can also be harmonised between scientifically closely related INSPIRE themes like the Atmospheric Conditions & Meteorological Geographical Features and the Oceanographic Geographical Features. The O&M model is very good basis for such harmonisation within the INSPIRE themes dealing with scientific measurement and numerical prediction data sets.
The data models in O&M based INSPIRE themes take an event based perspective on the provided data sets: The main actors in the play are temporal Observation events, and a coverage data is only one of the properties of the Observation instance. In addition to this result property, the properties for the feature of interest, the observed property, the used procedure and other metadata are provided with each OM_Observation instance. On the other hand most existing binary data formats for storing scientific data sets are (grid) coverage oriented: they are designed to efficiently store and access gridded coverage data, and currently provide only limited means of encoding the other OM_Observation properties within their internal data structure. For this reason most of the currently existing scientific binary data formats do not fit very well to encoding the the O&M based INSPIRE data sets using the in-band result delivery. Even if technically possible, modifying large sets of existing data files by embedding the missing OM_Observation properties would be costly or at least impractical.

**B.3 Returning OM_Observation results in-band or out-of-band**

There are several different technical options for encoding the OM_Observation grid valued data. The most straight-forward options are using in-band result delivery methods with GML encoding. This option is also closest to the existing delivery mechanisms for INSPIRE Annex I themes. For large grid coverage data sets the practical limits of the XML encoding considering data transfer and data access efficiency are soon reached with these options.

The other end of the spectrum would be to use pure binary O&M in-band encodings, which would simplify the life of those users capable of decoding those formats, but on the other hand be completely inaccessible for the users only capable of understanding GML encoded data. There is ongoing work at the OGC of defining the CF-NetCDF format and data model for encoding O&M Observation data, but generally such binary data formats are not common.

The out-of-band result delivery can be seen as a middle way approach to this problem: all users capable of understanding GML encoded OM_Observations are able to get a good deal of information for deciding whether the data set is interesting enough for their use cases: the feature of interest and the sampling feature describe the geographical location of the target of the Observation, the observed property gives information on what aspects of the target have been observed, the process and the metadata can be used for finding out the details of the Observation event, and the phenomenonTime property of the OM_Observation places the event at a point in the time axis. After this initial evaluation step the users may decide to proceed with the possibly resource and time consuming operation of downloading the separately available result grid coverage.

This section presents some of the options for encoding the value of grid coverage valued result properties of OM_Observations. Both in-band and out-of-band options are included as the most feasible choice depends on the quality and the quantity of the delivered data sets.

For each example, only the om:result element of the OM_Observation instance if provided for brevity.

**B.3.1 Option 1 (in-band): Embed the result using GML Coverage encoding**

In this case a gmlcov:ReferenceableGridCoverage is used, but a gml:RectifiedGridCoverage could also be appropriate. The encoding shows an x,y,z t grid coverage

```xml
<om:result>
  <gmlcov:ReferenceableGridCoverage gml:id="refgrid1">
    <gml:domainSet>
      <gml:ReferenceableGridByVectors dimension="3" gml:id="refgridv1" srsName="urn:ogc:def:crs:EPSG:6.6:4326 ++timeCRS definition++">
        <gml:limits>
          <gml:GridEnvelope>
            <gml:low>0 0 0</gml:low>
            <gml:high>180 359 365</gml:high>
          </gml:GridEnvelope>
        </gml:limits>
      </gml:ReferenceableGridByVectors>
    </gml:domainSet>
  </gmlcov:ReferenceableGridCoverage>
</om:result>
```
For several OM_Observation instances with the same domainSet and rangeType (like periodical new observations), the recurring parts can be re-used using XLink syntax. This makes the om:result element much more compact:
Pros:

- Uses OGC standard coverage encodings.
- The domain, and rangeType are fully encoded.

Cons:

- Encoding the rangeSet inline is not suitable for very large datasets.

**B.3.2 Option 2: (out-of-band) XLink the entire om:result**

The idea is to use the “GML way” of linking to an external resource: the XLink.

**Example:**

```
<om:result
   xlink:type="simple"
   xlink:href="http://web.access.server.org/pre-defined/cv-12345.nc"/>
```

Pros:

- Very simple to implement.

Cons:

- No way to define MIME type or the size of the remote file in advance, client must do “trial-and-error” by making a HTTP HEAD request first for example.
- Offers practically no information about the result (e.g. the domain of the coverage), clients unable to represent a geospatial placeholder area or point before downloading the entire binary file.
B.3.3 Option 3 (out-of-band): Define an INSPIRE specific referencing “out-of-band” result type

This option is semantically just an extended version of the XLink option (3), but with possibility to properly define the service interfaces.

Example:

```
<om:result>
  <inspire_om:OutOfBandResult>
    <inspire_om:AlternativeAccessMethods>
      <inspire_om:OnlineAccessService>
        <inspire_om:serviceAddress>http://data.access.server.org/wcs</inspire_om:serviceAddress>
        <inspire_om:protocol>http://www.opengis.net/spec/WCS/2.0</inspire_om:protocol>
        <inspire_om:dataIdentifier>cv-12345</inspire_om:dataIdentifier>
        <inspire_om:queryMethod>GetCoverage</inspire_om:queryMethod>
      </inspire_om:OnlineAccessService>
    </inspire_om:AlternativeAccessMethods>
  </inspire_om:OutOfBandResult>
</om:result>
```

Pros:

- Can be used with no model conflicts for those OM_Observation types that do not restrict the result type.
- Possibility to give alternative access methods for a result coverage.

Cons:

- Does not comply to the current version of SamplingCoverageObservation (or the EnvironmentalSamplingObservation or what ever it will be called in the final INSPIRE OM model): the om:result OutOfBandResult is not of type DiscreteCoverage, would need to be changed to allow either a DiscreteCoverage or an OutOfBandResult
- Mixes the the model content and the network service semantics, but on the other hand, so does any of the proposed out-of-band options.
- INSPIRE specific solution, no existing community support.
- Offers very little information about the result (e.g. the domain of the coverage).
B.3.4 Option 4 (out-of-band): Link to an external rangeSet using gml:File

This option tries to follow the existing methods in GML for referencing external resources (gml:File), but it also tries to "slip in" the possibility for knowledgeable clients to use alternative methods for accessing the coverage range data or a subset of it.

Example:

```xml
<om:result>
  <gmlcov:ReferenceableGridCoverage gml:id="refgrid1">
    <gml:domainSet>
      <csml:ReferenceableGridByVectors dimension="3" gml:id="refgridv1"
        srsName="urn:ogc:def:crs:EPSG:6.6:4326 ++timeCRS definition++">
        <gml:limits>
          <gml:GridEnvelope>
            <gml:low>0 0 0</gml:low>
            <gml:high>180 359 365</gml:high>
          </gml:GridEnvelope>
          <gml:axisLabels>x y t</gml:axisLabels>
          <origin>
            <gml:Point gml:id="originID">
              <!-- the srs of this point is the srs of the external reference system i.e. the srs specified in the ReferenceableGridByVectors srsName attribute -->
              <gml:pos>0 0 0</gml:pos>
            </gml:Point>
          </origin>
        </gml:limits>
        <generalGridAxis>
          <GeneralGridAxis>
            <offsetVector>1 0 0</offsetVector>
            <coefficients>-90.0 -89.0 -88.0.... 88.0 89.0 90</coefficients>
            <gridAxesSpanned>x y</gridAxesSpanned>
            <sequenceRule axisOrder="+1 +2">Linear</sequenceRule>
          </GeneralGridAxis>
        </generalGridAxis>
        <generalGridAxis>
          <GeneralGridAxis>
            <offsetVector>0 1 0</offsetVector>
            <coefficients>-180.0 -179.0 -178.0 ....178.0 179.0 180.0</coefficients>
            <gridAxesSpanned>x y</gridAxesSpanned>
            <sequenceRule axisOrder="+1 +2">Linear</sequenceRule>
          </GeneralGridAxis>
        </generalGridAxis>
        <generalGridAxis>
          <GeneralGridAxis>
            <offsetVector>0 0 1</offsetVector>
            <!-- coefficients correspond to "n days since 2010-01-01T00:00:00.0"
             (the temporal crs)--> 
            <coefficients>0 1 2 3 4 5 ... 362 363 364</coefficients>
            <gridAxesSpanned>t</gridAxesSpanned>
            <sequenceRule axisOrder="+1">Linear</sequenceRule>
          </GeneralGridAxis>
        </generalGridAxis>
      </csml:ReferenceableGridByVectors>
    </gml:domainSet>
  </gmlcov:ReferenceableGridCoverage>
</om:result>
```

```xml
<inspire_om:ExternalStorageDescriptor>
  <inspire_om:descriptor>
    <inspire_om:NetCDFExtract>
      <inspire_om:variableName>air_temperature</inspire_om:variableName>
    </inspire_om:NetCDFExtract>
  </inspire_om:descriptor>
</inspire_om:ExternalStorageDescriptor>
```
The ExternalStorageDescriptor element placed within the gml:rangeParameters element is almost identical to the CSMLStorageDescriptor element. The abstract FileExtractType is however simplified for the out-of-band use by leaving out the mandatory arraySize and fileName properties, as well as the optional uom and numericType properties used for defining embedded numeric data array structures inherited in CSML from the abstract ArrayDescriptorType. The NetCDFExtract element in the example above tells the client that only the air_temperature variable values within the referred NetCDF file form the rangeSet of this coverage.

Besides the NetCDFExtract, there is also a GRIBExtract element defined in CSML. This is also copied to the XML schema of Annex I.

As with option 1, several OM_Observation instances with the same domainSet and rangeType (like periodical new observations), could re-use these elements for more compact encoding:

```xml
<om:result>
  <gmlcov:ReferenceableGridCoverage gml:id="refgrid1" xlink:href="http://web.access.server.org/pre-defined/cv-23443-timeseries.domainset.xml">
    <gml:domainSet xlink:href="http://web.access.server.org/pre-defined/cv-23443-timeseries.domainset.xml">
      <gml:rangeSet>
        <gml:File>
          <gml:rangeParameters>
            <inspire_om:ExternalStorageDescriptor>
              <inspire_om:descriptor>
                <inspire_om:NetCDFExtract>
                  <inspire_om:variableName>air_temperature</inspire_om:variableName>
                </inspire_om:NetCDFExtract>
              </inspire_om:descriptor>
            </inspire_om:ExternalStorageDescriptor>
          </gml:rangeParameters>
        </gml:File>
      </gml:rangeSet>
    </gml:domainSet>
  </gmlcov:ReferenceableGridCoverage>
</om:result>
```
Pros:

- We don’t break the current model: om:result for the SamplingGridCoverage is of type DiscreteCoverage.
- Follows the GML standard GML coverage encoding model.
- Even if the clients cannot understand the rangeParameters, they are able to access the entire binary file using the fileReference and mimeType properties (provided that they understand the gml:File).

Cons:

- The external binary files typically contain the entire coverage description, not just the rangeSet. Actually we should probably point inside the binary files in gml:File to access the rangeSet directly. This kind of internal references are often file format specific, if they exist at all.
- Possibly slight misuse of gml:rangeParameters: the GML schema say it should be used for “semantics of the range set”.

B.3.5 Option 5 (out-of-band): gml:File with AlternativeEncodings

This is option is very similar to the option 4, but it adds possibility to provide alternative access methods to retrieve the coverage data. The elements used for this are the same as in option 3, but this time they are provided within the gml:rangeParameters.

Example:

```xml
<om:result>
  <gmlcov:ReferenceableGridCoverage gml:id="refgrid1">
    <gml:domainSet xlink:href="http://web.access.server.org/pre-defined/cv-2343-timeseries.domainset.xml">
      <gml:rangeSet>
        <gml:File>
          <gml:rangeParameters>
            <inspire_om:AlternativeAccessMethods>
              <inspire_om:OnlineAccessService>
                <inspire_om:serviceAddress>http://data.access.server.org/wcs</inspire_om:serviceAddress>
                <inspire_om:protocol>http://www.opengis.net/spec/WCS/2.0</inspire_om:protocol>
                <inspire_om:queryMethod>GetCoverage</inspire_om:queryMethod>
              </inspire_om:OnlineAccessService>
              <inspire_om:OnlineAccessService>
                <inspire_om:serviceAddress>ftp://ftp.access.server.org</inspire_om:serviceAddress>
                <inspire_om:protocol>FTP</inspire_om:protocol>
                <inspire_om:queryMethod>RETR</inspire_om:queryMethod>
              </inspire_om:OnlineAccessService>
              <inspire_om:OnlineAccessService>
                <inspire_om:serviceAddress>http://web.access.server.org</inspire_om:serviceAddress>
                <inspire_om:protocol>HTTP</inspire_om:protocol>
                <inspire_om:queryMethod>GET+NetCDFExtract</inspire_om:queryMethod>
              </inspire_om:OnlineAccessService>
            </inspire_om:AlternativeAccessMethods>
          </gml:rangeParameters>
        </gml:File>
      </gml:rangeSet>
    </gml:domainSet>
  </gmlcov:ReferenceableGridCoverage>
</om:result>
```
<inspire_om:variableName>air_temperature</inspire_om:variableName>
</inspire_om:NetCDFExtract>
</inspire_om:descriptor>
</inspire_om:ExternalStorageDescriptor>
</inspire_om:accessParameters>
</inspire_om:OnlineAccessService>
</inspire_om:AlternativeAccessMethods>
</gml:rangeParameters>
<gml:fileReference>http://web.access.server.org/pre-defined/cv-12345.nc</gml:fileReference>
<gml:fileStructure codeSpace="cf-conventions-version">1.6</gml:fileStructure>
<gml:mimeType>application/x-netcdf</gml:mimeType>
</gml:File>
</gml:rangeSet>
<gmlcov:rangeType xlink:href="http://web.access.server.org/pre-defined/cv-23443-timeseries.rangetype.xml"/>
</gmlcov:ReferenceableGridCoverage>
</om:result>

Pros:

- Follows standard GML coverage encoding
- Possibility to give alternative access methods for a result coverage, including service interfaces for returning only part of the entire coverage based on users’ requirements.

Cons:

- A bit of a hack to provide references to alternative access methods inside the gml:File element: those access methods should really be one level up, but there does not seem to be a an appropriate extension point in GML at that level.

B.3.6 Option 6 (out-of-band): Using SWE Common DataStream & BinaryEncoding

The SWE Common also has means of referring to external, binary encoded data arrays. In the following example the value of the om:result is a swe:DataStream representing a 3000 by 3000 pixel raster image, with colors of each pixel encoded using one unsigned byte for each three components: red, green and blue:

Example:

<om:result>
<swe:DataStream id="EXAMPLE_03">
<!-- -->
<swe:elementType name="imageData">
<swe:DataArray definition="http://sweet.jpl.nasa.gov/2.0/info.owl#Raster">
<swe:elementCount>
<swe:Count>
<swe:value>3000</swe:value>
</swe:Count>
</swe:elementCount>
<swe:elementType name="row">
<swe:DataArray definition="http://sweet.jpl.nasa.gov/2.0/info.owl#Row">
<swe:elementCount>
<swe:Count>
<swe:value>3000</swe:value>
</swe:Count>
</swe:elementCount>
<swe:elementType name="pixel">
<swe:DataRecord definition="http://sweet.jpl.nasa.gov/2.0/info.owl#Cell">
  <swe:field name="red">
    <swe:Count definition="http://sweet.jpl.nasa.gov/2.0/physRadiation.owl#Radiance">
      <swe:description>
        Radiance measured on band1 usually assigned to red channel
      </swe:description>
      <swe:nilValues id="NIL_VALUES">
        <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/BelowDetectionRange">0</swe:nilValue>
        <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange">255</swe:nilValue>
      </swe:nilValues>
    </swe:Count>
  </swe:field>
  <swe:field name="green">
    <swe:Count definition="http://sweet.jpl.nasa.gov/2.0/physRadiation.owl#Radiance">
      <swe:description>
        Radiance measured on band2 usually assigned to green channel
      </swe:description>
      <swe:nilValues xlink:href="#NIL_VALUES"/>
    </swe:Count>
  </swe:field>
  <swe:field name="blue">
    <swe:Count definition="http://sweet.jpl.nasa.gov/2.0/physRadiation.owl#Radiance">
      <swe:description>
        Radiance measured on band3 usually assigned to blue channel
      </swe:description>
      <swe:nilValues xlink:href="#NIL_VALUES"/>
    </swe:Count>
  </swe:field>
</swe:DataRecord>
Pros:

- SWE Common is an OGC standard and widely used, no need to create an INSPIRE specific solution.
- The structure of the referenced data array is well defined in the GML encoding, so the clients know what to expect when downloading the binary encoded part.

Cons:

- The om:result does not define a coverage: there is not explicit definition of the spatial and/or temporal locations of the grid points. The result is just an array of data cells. The clients are not able to map the values to a geospatial area and/or on the time axis without added information.

B.3.7 Option 7 CSML Pattern

TBD

B.4 XML Schema definitions for the proposed out-of-band result types

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified"
targetNamespace="http://inspire.jrc.ec.europa.eu/schemas/om_common/3.0">
  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd"/>

  <element name="OnlineAccessService" type="inspire_om:OnlineAccessServiceType"/>

  <complexType name="OnlineAccessServiceType">
    <sequence>
      <element name="serviceAddress" type="anyURI"/>
      <element name="protocol" type="anyURI"/>
      <element name="dataIdentifier" type="anyURI"/>
      <element name="queryMethod" type="anyURI" minOccurs="0"/>
      <element name="accessParameters" type="gml:AssociationRoleType" minOccurs="0"/>
    </sequence>
  </complexType>

  <element name="AlternativeAccessMethods" type="inspire_om:AlternativeAccessMethodArrayType"/>

  <complexType name="AlternativeAccessMethodArrayType">
    <sequence>
      <element ref="inspire_om:OnlineAccessService" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <element name="OutOfBandResult" type="inspire_om:OutOfBandResultType"/>
</schema>
```
<complexType name="OutOfBandResultType">
  <sequence>
    <element ref="inspire_om:AlternativeAccessMethods"/>
  </sequence>
</complexType>

<!------------------------------------------------------------------------------->
<!------ Root storage descriptor element  ------>
<!------ Modeled closely after the CSMLStorageDescriptor ------  ----->
<complexType name="ExternalStorageDescriptorType">
  <sequence>
    <element name="descriptor" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element ref="inspire_om:FileExtract"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>

<element name="ExternalStorageDescriptor" type="inspire_om:ExternalStorageDescriptorType"/>
<complexType name="CSMLStorageDescriptorPropertyType">
  <sequence>
    <element ref="inspire_om:ExternalStorageDescriptor"/>
  </sequence>
</complexType>

<!------------------------------------------------------------------------------->
<complexType name="FileExtractType" abstract="true">
  <annotation>
    <documentation>
    </documentation>
  </annotation>
</complexType>

<element name="FileExtract" type="inspire_om:FileExtractType" abstract="true"/>

<!------------------------------------------------------------------------------->
<complexType name="NetCDFExtractType">
  <complexContent>
    <extension base="inspire_om:FileExtractType">
      <sequence>
        <element name="variableName" type="string"/>
        <element name="recordNumber" type="int" minOccurs="0"/>
        <element name="fileOffset" type="int" minOccurs="0"/>
        <element name="ctlVariableName" type="string" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="NetCDFExtract" type="inspire_om:NetCDFExtractType" substitutionGroup="inspire_om:FileExtract"/>
<complexType name="NetCDFExtractPropertyType">
  <sequence>
    <element ref="inspire_om:NetCDFExtract"/>
  </sequence>
</complexType>

<!------------------------------------------------------------------------------->
<complexType name="GRIBExtractType">
  <complexContent>
    <extension base="inspire_om:FileExtractType">
      <sequence>
        <element name="parameterCode" type="int"/>
        <element name="recordNumber" type="int" minOccurs="0"/>
        <element name="fileOffset" type="int" minOccurs="0"/>
        <element name="ctlVariableName" type="string" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</complexType>
<element name="GRIBExtract" type="inspire_om:GRIBExtractType"
substitutionGroup="inspire_om:FileExtract"/>
<complexType name="GRIBExtractPropertyType">
<sequence>
<element ref="inspire_om:GRIBExtract"/>
</sequence>
</complexType>
</schema>