Lightweight Sensor Web Enablement Profile for Stationary In-Situ Sensors

(Version 2011-02-10)

Simon Jirka (jirka@52north.org)
Christoph Stasch (staschc@uni-muenster.de)

Introduction

The OGC Sensor Web Enablement architecture (SWE) addresses the integration of sensors and sensor data into Spatial Data Infrastructures (SDI). Within the SWE architecture the Sensor Observation Service (SOS) plays a central role as it defines an interface for accessing sensor data and metadata. This document describes a lightweight profile of the SOS and the data formats used by the SOS: Observations & Measurements (O&M) for encoding measurement data and the Sensor Model Language (SensorML) for encoding metadata. Other SWE standards which provide more specialized functionality are not part of this minimum lightweight SWE profile.

This profile has been designed in a way that is on the one hand efficient and easy to implement and on the other hand standard compliant. Especially the following aspects were considered during the creation of the profile:

- Reducing the number of operations: certain operations of the SOS standard were designed for very specific needs; these operations were left out of this profile
- Reducing the complexity of the SOS operations (e.g. less complex filters for requesting sensor data)
- Limit the supported data formats to those used in practice (i.e. no support of TML within the SOS)
- Focus on fixed in-situ sensors as these sensors are the type of sensor that is used in the broad majority of SWE use cases in practice (for the future an extension may become necessary)

In summary the main objective that guided the development of this profile was to support those use cases which are regularly occurring in practice and to leave out those with very specific requirements that go beyond the broad mass of SWE use cases.

In the next section it will be described, how a lightweight profile of the SOS should look like. At the end of this document an annex is provided that contains examples of the operations and encodings that are part of this profile.

SOS Light

The OGC Sensor Observation Service is a standardized interface for accessing sensor data and metadata. Thus, the SOS can be considered the core element of the SWE architecture. It provides a means for integrating sensor data into Spatial Data Infrastructures like it is already possible for other kinds of geospatial data such as maps (OGC Web Map Service), raster data (OGC Web Coverage Service) and geometry objects (OGC Web Feature Service).

The following subsections will explain which parts of the SOS need to be considered for a minimum profile that reduces its complexity and increases the efficiency.

SOS Operations

This lightweight SOS profile comprises those operations which are mandatory within the SOS standard (GetCapabilities, DescribeSensor and GetObservation) so that they must be implemented by every standard compliant SOS server. Furthermore the GetFeatureOfInterest operation has been included as it allows accessing the geometries of sensor stations.

GetCapabilities

The GetCapabilities operation is common for every OGC web service specification. It is important to support this operation as it provides the metadata necessary for clients to generate valid SOS requests.

It contains especially the following information:

- Supported operations of the SOS instance
- Spatial extent for which data is available
- Identifiers of all sensor instances encapsulated by the SOS instance
- Identifiers of all geometric features for which sensor data is available
- Identifiers of all observed phenomena
- Offerings (the concept of an offering can be compared to a layer; within an offering it is possible to group observation data that concerns a common spatial extent, a common time period and/or a common thematic)

DescribeSensor

The DescribeSensor operation allows retrieving metadata about the sensors encapsulated by a SOS instance. It is a mandatory element of the SOS specification. Thus it is included in the minimum profile so that servers must offer this operation. Clients however may not necessarily make use of the DescribeSensor operation.

The request of the DescribeSensor operation is very simple as the only parameter consists of the identifier of the sensor for which a metadata document is requested. The response is a SensorML document (a simple structure for such a SensorML document is defined below).

GetObservation

The GetObservation operation is an essential part of the SOS specification as it is the operation for requesting sensor data. Thus, it is obvious that this operation is needed for both servers and clients. However, in order to reduce the complexity of the GetObservation operation several restrictions have been defined that limit the query parameters. In detail the following parameters are included in this lightweight SOS profile:

- Offerings (required by the SOS standard): the identifier of the offering that contains the requested sensor data
- Temporal filters for specifying the time for which sensor data is requested; however the temporal filters are limited within the lightweight SOS profile to the following types:
    - During a time period
    - Equal to a time instant
    - After a time instant
    - Before a time instant
- Procedure: the identifiers of the sensor instances from which data is requested
- Observed property: the identifiers of the phenomena for which data is requested
- Feature of interest: the geometry object for which data is requested; within the lightweight SOS profile this is limited to:

    - Identifier of the geometric objects for which sensor data is requested
    - Bounding box of an area for which data is requested
- Result: Constraints on the values of the sensor data that is requested; within the lightweight SOS profile this is limited to:
    - For numeric values:
        - Between
        - Equal to
        - Not equal to
        - Less than
        - Less than or equal to
        - Greater than
        - Greater than equal to
    - For string values:
        - Equal to
        - Like
- Furthermore the following three parameters must be provided. However they are restricted within the lightweight SOS profile to fixed values:
    - Response format: the data format in which the response shall be encoded; fixed value: “text/XML subtype=”OM\1.0.0”” (for a restriction of O&M see below)
    - Result model: the way how the sensor data shall be encoded; currently we propose to select one of the two options (for more information please see the O&M section):
        - om:observation ensures a compact encoding so that multiple sensor data values can be encoded as comma separated values avoiding multiple redundant XML elements; the result shall be a swe:DataArray; the cost for decoding of the observed values is higher compared to om:measurement
        - om:measurement allows an easier decoding of measured values as it completely relies on a XML structure; at the same time this leads to a higher volume of data that is transmitted
    - Response mode: value indicating if the sensor data shall be provided directly within the O&M response or if it shall contain pointers to external references; fixed value: “inline” (this ensures, that all measurements are directly contained in the O&M response)

GetFeatureOfInterest

The GetFeatureOfInterest operation allows retrieving the geometries of the stations at which measurements were performed. As the access to the geometries of measurements is a requirement for creating map visualizations of the sensors and their data, the GetFeatureOfInterest operation has been included for servers in this lightweight SOS profile. However, not every client might rely on this operation (e.g. clients that only show diagrams of time series data).

In order to reduce the complexity of this operation, its request parameters are restricted within this profile. Clients may request geometries using only one of these two parameters:

- the identifiers of the features of interest
- a bounding box describing an area of interest

Observations & Measurements

The Observations & Measurements standard offers an encoding for data observed by sensors. However, as O&M has been designed to be extremely versatile and comprehensive it has also a certain complexity. Thus, this lightweight profile has been defined containing only those elements of O&M which are essential for most common use cases. Within this document we describe two different observation types which differ by the result models used: “om:Observation” and “om:Measurement”.

Observation

In order to minimize the data volume for transmitting sensor data, the responses of the SOS can be transmitted as om:Observation which allow a more compact encoding of sensor data, as multiple values (and timestamps) can efficiently be encoded in one observation using an approach like comma separated values. The drawback of this approach is a more complex decoding mechanism for the comma separated encoding of the measured values. In order to reduce the complexity of O&M Observations, only a limited set of O&M elements shall be used. These are:

- Sampling Time (mandatory): this element describes the time instant or time period for which the observation contains sensor data; within the lightweight profile this is restricted to the data types TimeInstant and TimePeriod (other kinds of time (result and valid time) are currently not considered for this profile)
- Procedure (mandatory): the identifier of the sensor instance that has generated the observation
- Observed property (mandatory): the identifier of the phenomenon that was observed
- Feature of interest (mandatory): an identifier of the geometric feature (e.g. sensor station) to which the observation is associated; within the lightweight profile this is limited to sampling points
- Result (mandatory): the measured values; in addition to the restriction that only om:Observations shall be supported, we limit the result type to SWE Data Array; such a SWE Data Array consists of records with time stamps and values which are text block encoded (with specific delimiters between the records and fields)
- Quality (optional): name value pairs describing relevant quality parameters of the observation

Measurement

Using the om:Measurement result model, the decoding of the measured values is facilitated as the parsing of the response can be completely based on the XML structure. However this approach has the drawback of a higher data volume, as each measurement is encoded within a separate XML element.

Also in this case the SWE lightweight profile limits the occurring elements within O&M documents:

- Sampling Time (mandatory): this element describes the time instant or time period for which the observation contains sensor data; within the lightweight profile this is restricted to the data types TimeInstant and TimePeriod (other kinds of time (result and valid time are currently not considered for this profile)
- Procedure (mandatory): the identifier of the sensor instance that has generated the observation
- Observed property (mandatory): the identifier of the phenomenon that was observed

- Feature of interest (mandatory): an identifier of the geometric feature (e.g. sensor station) to which the observation is associated; within the lightweight profile this is limited to sampling points
- Result (mandatory): the measured values, including the unit of measurement
- Quality (optional): name value pairs describing relevant quality parameters of the observation

Sensor Model Language

The Sensor Model Language (SensorML) is used within the SOS for encoding sensor metadata documents that are returned in case of DescribeSensor requests.

This lightweight profile defines a minimum set of metadata that shall be provided in a SensorML document. Complex elements of SensorML that are intended for very specific applications are not considered within this lightweight profile but may be added if necessary.

The following metadata items are considered mandatory for a SensorML document:

- Identifiers of the sensor instances (a unique identifier, a short name and a long name)
- A classifier identifying the sensor type
- Contact information about the operator of the sensor
- The position of the sensor
- The outputs of the sensors (identifiers of the observed phenomena and if necessary the units of measurement)

A further restriction is that every sensor shall be modeled as a SensorML system. Other types are not considered for this profile.

Outlook

Currently the SOS 2.0 specification is in the specification process at the OGC. As the SOS 2.0 is expected to be a rather evolutionary approach than a completely new interface, this lightweight SOS profile will remain mostly valid. However, for the SOS 2.0 a RESTful binding will be included in the standard, which might provide an additional means for accessing SOS servers.

Examples

The following examples illustrate how the previously described operations and encodings would look like in practice.

SOS Operations

DescribeSensor

This example shows a DescribeSensor request for accessing the metadata of the sensor with the identifier “urn:x-ogc:object:sensor:airquality:10002”.

```
<DescribeSensor>

   <!-- the identifier of the sensor for which a metadata document is requested -->
   <procedure>urn:x-ogc:object:sensor:airquality:10002</procedure>

</DescribeSensor>
```

The response to the DescribeSensor operation is shown below in the example on SensorML.

GetObservation

The following four examples show typical cases of GetObservation-Requests.

The following message is used for requesting all ozone data measured by the sensor with the identifier “urn:x-ogc:object:sensor:airquality:10002”.

Depending on the selected result model the content of the resultModelElement may either be “om:Observation” or “om:Measurement”.

```
<GetObservation>

   <!-- the offering of the SOS that contains the requested data -->
   <offering>Ozone</offering>

   <!-- the identifier of the sensor from which data is requested -->
   <procedure>urn:x-ogc:object:sensor:airquality:10002</procedure>

   <!-- the phenomenon about which data is requested -->
   <observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</observedProperty>

   <!-- the requested response format and result model -->
   <responseFormat>text/xml;subtype=&amp;quot;om/1.0.0&amp;quot;</responseFormat>
   <resultModel>om:Measurement</resultModel>

</GetObservation>
```

The next example shows a request for all ozone data that was measured during a user defined time period.

```
<GetObservation>

   <!-- the offering of the SOS that contains the requested data -->
   <offering>Ozone</offering>
   <eventTime>
      <ogc:TM_During>
         <ogc:PropertyName>om:samplingTime</ogc:PropertyName>

         <!-- the time period for which ozone data is requested; this is defined by start and
         end time of the period-->
         <gml:TimePeriod>
            <gml:beginPosition>2010-06-01T00:00:00+00:00</gml:beginPosition>
            <gml:endPosition>2010-06-03T23:59:59+00:00</gml:endPosition>
         </gml:TimePeriod>
      </ogc:TM_During>
   </eventTime>

   <!-- the phenomenon about which data is requested -->
   <observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</observedProperty>

   <!-- the requested response format and result model -->
   <responseFormat>text/xml;subtype=&quot;om/1.0.0&quot;</responseFormat>
   <resultModel>om:Measurement</resultModel>

</GetObservation>
```

This example shows a request for all ozone measurements with values greater than or equal to 125.

```
<GetObservation>

   <!-- the offering of the SOS that contains the requested data -->
   <offering>Ozone</offering>

   <!-- the phenomenon about which data is requested -->
   <observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</observedProperty>
   <result>

      <!-- the filter specifying the greater than or equal to constraint (in this case: all
      ozone values greater than or equal to 125 -->
      <ogc:PropertyIsGreaterThanOrEqualTo>
         <ogc:PropertyName>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</ogc:PropertyName>
            <ogc:Literal>125</ogc:Literal>
         </ogc:PropertyIsGreaterThanOrEqualTo>
   </result>

   <!-- the requested response format and result model-->
   <responseFormat>text/xml;subtype=&quot;om/1.0.0&quot;</responseFormat>
   <resultModel>om:Measurement</resultModel>

</GetObservation>
```

This example shows a request for all ozone measurements with values between 100 and 125.

```
<GetObservation>

   <!-- the offering of the SOS that contains the requested data -->
   <offering>Ozone</offering>

   <!-- the phenomenon about which data is requested -->
   <observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</observedProperty>

   <!-- the filter specifying the interval in which the ozone measurements shall lie -->
   <result>
      <ogc:PropertyIsBetween>
         <ogc:PropertyName>urn:ogc:def:phenomenon:OGC:1.0.30:Ozone</ogc:PropertyName>

            <!-- the lower boundary -->
            <ogc:LowerBoundary>
               <ogc:Literal>100</ogc:Literal>
            </ogc:LowerBoundary>

            <!-- the upper boundary -->
            <ogc:UpperBoundary>
               <ogc:Literal>125</ogc:Literal>
            </ogc:UpperBoundary>
         </ogc:PropertyIsBetween>
      </result>

   <!-- the requested response format and result model -->
   <responseFormat>text/xml;subtype=&quot;om/1.0.0&quot;</responseFormat>
   <resultModel>om:Measurement</resultModel>

</GetObservation>
```

The response to the GetObservation operation is shown below in the example on Observations & Measurements.

GetFeatureOfInterest

The following request is used for requesting the geometry of the feature with the identifier "foi_10002".

```
<GetFeatureOfInterest>

   <!-- identifier of the requested feature -->
   <FeatureOfInterestId>foi_10002</FeatureOfInterestId>

</GetFeatureOfInterest>
```

The next example is instead used for requesting the geometry of all features within a given bounding box.

```
<GetFeatureOfInterest>
   <location>

      <!-- bounding box for which the feature geometries are requested -->
      <ogc:BBOX>
         <ogc:PropertyName>urn:ogc:data:location</ogc:PropertyName>

         <!-- the description of the bounding box including the identifier of the reference
         system and the coordinates of the lower and upper corner -->
         <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
            <gml:lowerCorner>45.0 7.0</gml:lowerCorner>
            <gml:upperCorner>53.0 17.0</gml:upperCorner>
         </gml:Envelope>

      </ogc:BBOX>
   </location>

</GetFeatureOfInterest>
```

Below an example of a response to a GetFeatureOfInterest request is shown. In this case it contains one sampling point.

```
<gml:FeatureCollection>
   <gml:featureMember>

      <!-- the identifier of the sampling point -->
      <sa:SamplingPoint gml:id="foi_10002">

         <!-- the name of the sampling point -->
         <gml:name>10002</gml:name>

         <!-- the position of the sampling point -->
         <sa:position>
            <gml:Point>
               <gml:pos srsName="urn:ogc:def:crs:EPSG:4326">47.299999 16.193064</gml:pos>
            </gml:Point>
         </sa:position>

      </sa:SamplingPoint>

   </gml:featureMember>
</gml:FeatureCollection>
```

Observations & Measurements

Example 1: Result Model "om:Observation"

The example below shows an O&M observation returned by a GetObservation request using the result model "om:Observation". In this example a series of 17 ozone measurements is contained.

```
<om:Observation>

   <!-- time period for which this O&M document contains observations -->
   <om:samplingTime>
      <gml:TimePeriod xsi:type="gml:TimePeriodType">
         <gml:beginPosition>2010-06-02T06:00:00.000+02:00</gml:beginPosition>
         <gml:endPosition>2010-06-07T08:00:00.000+02:00</gml:endPosition>
      </gml:TimePeriod>
   </om:samplingTime>

   <!-- identifiers of the sensors that have created the observations contained in
   this document -->
   <om:procedure xlink:href="http://x-test.myserver.org/sensors/10002"/>

   <!-- description of the observed phenomenon -->
   <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone"/>

   <!-- description of the sensor stations that generated the observations -->
   <om:featureOfInterest xlink:href="http://x-test.myserver.org/features/10002"/>

   <!-- the observation results encoded as a data array -->
   <om:result>
      <swe:DataArray>

         <!-- description of the fields of this data array: in this case time, feature
         (sensor station) and value -->
         <swe:elementType>
            <swe:SimpleDataRecord>
               <swe:field name="Time">
                  <swe:Time definition="urn:ogc:data:time:iso8601"/>
               </swe:field>
               <swe:field name="feature">
                  <swe:Text definition="urn:ogc:data:feature"/>
               </swe:field>
               <swe:field name="Ozone">
                  <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone">
                     <swe:uom code="µg/m^3"/>
                  </swe:Quantity>
               </swe:field>
            </swe:SimpleDataRecord>
         </swe:elementType>

         <!-- description how the values are encoded (definition of the used separators
         for records and fields) -->
```

```
         <swe:encoding>
            <swe:TextBlock decimalSeparator="." tokenSeparator="," blockSeparator=";"/>
         </swe:encoding>

         <!-- the values of the observations -->
         <swe:values>
            2010-06-02T06:00:00.000+02:00,foi_10002,30;
            2010-06-02T07:00:00.000+02:00,foi_10002,44;
            2010-06-02T09:00:00.000+02:00,foi_10002,40;
            2010-06-02T12:00:00.000+02:00,foi_10002,46;
            2010-06-05T04:00:00.000+02:00,foi_10002,33;
            2010-06-05T05:00:00.000+02:00,foi_10002,45;
            2010-06-05T06:00:00.000+02:00,foi_10002,57;
            2010-06-05T07:00:00.000+02:00,foi_10002,91;
            2010-06-05T08:00:00.000+02:00,foi_10002,96;
            2010-06-05T09:00:00.000+02:00,foi_10002,102;
            2010-06-05T10:00:00.000+02:00,foi_10002,111;
            2010-06-05T11:00:00.000+02:00,foi_10002,116;
            2010-06-07T04:00:00.000+02:00,foi_10002,16;
            2010-06-07T05:00:00.000+02:00,foi_10002,35;
            2010-06-07T06:00:00.000+02:00,foi_10002,58;
            2010-06-07T07:00:00.000+02:00,foi_10002,89;
            2010-06-07T08:00:00.000+02:00,foi_10002,97;
         </swe:values>

      </swe:DataArray>
   </om:result>
</om:Observation>
```

Example 2: Result Model "om:Measurement"

The example below shows an O&M observation returned by a GetObservation request using the result model "om:Measurement". In this example a series of 3 ozone measurements is contained.

```
<om:ObservationCollection>

   <!-- time bounding box for which this O&M document contains observations -->
   <gml:boundedBy>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
         <gml:lowerCorner>7.0 7.52</gml:lowerCorner>
         <gml:upperCorner>52.9 52.0</gml:upperCorner>
      </gml:Envelope>
   </gml:boundedBy>

   <!-- a series of om:member element which each contain one measurement -->
   <om:member>
      <om:Measurement gml:id="o_001">

         <!-- time period for which this O&M document contains observations -->
         <om:samplingTime>
            <gml:TimeInstant xsi:type="gml:TimeInstantType">
               <gml:timePosition>2010-06-02T06:00:00.000+02:00</gml:timePosition>
            </gml:TimeInstant>
         </om:samplingTime>

         <!-- identifiers of the sensors that have created the observations contained in this
              document -->
         <om:procedure xlink:href="http://x-test.myserver.org/sensors/10002"/>

         <!-- description of the observed phenomenon -->
         <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone"/>

         <!-- description of the sensor stations that generated the observations -->
         <om:featureOfInterest xlink:href="http://x-test.myserver.org/features/10002"/>

         <!-- the value of the measurement including the unit of measurement -->
         <om:result uom="µg/m^3">30</om:result>

      </om:Measurement>
   </om:member>

   <om:member>
      <om:Measurement gml:id="o_002">

      <!-- time period for which this O&M document contains observations -->
      <om:samplingTime>
            <gml:TimeInstant xsi:type="gml:TimeInstantType">
```

```
                <gml:timePosition>2010-06-02T07:00:00.000+02:00</gml:timePosition>
            </gml:TimeInstant>
         </om:samplingTime>

         <!-- identifiers of the sensors that have created the observations contained in this
           document -->
         <om:procedure xlink:href="http://x-test.myserver.org/sensors/10002"/>

         <!-- description of the observed phenomenon -->
         <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone"/>

         <!-- description of the sensor stations that generated the observations -->
         <om:featureOfInterest xlink:href="http://x-test.myserver.org/features/10002"/>

         <!-- the value of the measurement including the unit of measurement -->
         <om:result uom="µg/m^3">44</om:result>

      </om:Measurement>
   </om:member>

   <om:member>
      <om:Measurement gml:id="o_003">

         <!-- time period for which this O&M document contains observations -->
         <om:samplingTime>
            <gml:TimeInstant xsi:type="gml:TimeInstantType">
               <gml:timePosition>2010-06-02T09:00:00.000+02:00</gml:timePosition>
            </gml:TimeInstant>
         </om:samplingTime>

         <!-- identifiers of the sensors that have created the observations contained in this
              document -->
         <om:procedure xlink:href="http://x-test.myserver.org/sensors/10002"/>

         <!-- description of the observed phenomenon -->
         <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone"/>

         <!-- description of the sensor stations that generated the observations -->
         <om:featureOfInterest xlink:href="http://x-test.myserver.org/features/10002"/>

         <!-- the value of the measurement including the unit of measurement -->
         <om:result uom="µg/m^3">40</om:result>

      </om:Measurement>
   </om:member>

</om:ObservationCollection>
```

Sensor Model Language

The following example contains metadata of the sensor that has produced the ozone measurements shown above.

```
<sml:SensorML>
   <sml:member>
      <sml:System>
         <sml:identification>
            <sml:IdentifierList>

               <!-- The unique identifier of the sensor -->
               <sml:identifier>
                  <sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
                     <sml:value>urn:x-ogc:object:sensor:airquality:00002</sml:value>
                  </sml:Term>
               </sml:identifier>

               <!-- A long name for the sensor -->
               <identifier name="longName">
                  <Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
                     <value>Ozone sensor Muenster</value>
                  </Term>
               </identifier>
               <!-- A short name for the sensor -->
               <identifier name="shortName">
                  <Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
```

```
                  <value>00002</value>
               </Term>
            </identifier>
         </sml:IdentifierList>
      </sml:identification>
      <classification>
         <ClassifierList>

            <!-- a classifier indicating the type of the sensor station -->
            <classifier name="stationType">
               <Term definition="urn:ogc:def:classifier:OGC:1.0:stationType">
                  <value>traffic</value>
               </Term>
            </classifier>

            <!-- an optional classifier; in this case it indicates the type of the area
            around the station -->
            <classifier name="stationArea">
               <Term definition="urn:ogc:def:classifier:OGC:1.0:stationArea">
                  <value>urban</value>
               </Term>
            </classifier>
         </ClassifierList>
      </classification>

   <!-- contact details about the party which is responsible for the sensor -->
   <contact>
      <ResponsibleParty>

         <!-- the name of the contact person -->
         <individualName>Simon Jirka</individualName>

         <!-- the name of the organization -->
         <organizationName>52 North</organizationName>
         <contactInfo>

            <!-- the email address of the contact person -->
            <address>
               <electronicMailAddress>jirka@52north.org</electronicMailAddress>
            </address>

            <!-- the web site of the responsible organization -->
            <onlineResource xlink:href="http://www.52north.org"/>
         </contactInfo>
      </ResponsibleParty>
   </contact>

   <!-- the position of the sensor -->
   <sml:position name="sensorPosition">

      <!-- the reference system which is used for describing the sensor position-->
      <swe:Position referenceFrame="urn:ogc:def:crs:EPSG:4326">
         <swe:location>

            <!-- a vector containing the coordinates of the sensor location -->
            <swe:Vector gml:id="STATION_LOCATION">
               <swe:coordinate name="longitude">
                  <swe:Quantity axisID="x">
                     <swe:uom code="degree"/>
                     <swe:value>11.417789</swe:value>
                  </swe:Quantity>
               </swe:coordinate>
               <swe:coordinate name="latitude">
                  <swe:Quantity axisID="y">
                     <swe:uom code="degree"/>
                     <swe:value>47.276665</swe:value>
                  </swe:Quantity>
               </swe:coordinate>
               <swe:coordinate name="altitude">
                  <swe:Quantity axisID="z">
                     <swe:uom code="m"/>
                     <swe:value>570</swe:value>
                  </swe:Quantity>
               </swe:coordinate>
            </swe:Vector>
         </swe:location>
      </swe:Position>
   </sml:position>
```

```
      <!-- a list of the phenomena observed by the sensor; in this case Ozone -->
         <sml:outputs>
            <sml:OutputList>

               <!-- description of the output 'Ozone'-->
               <sml:output name="Ozone">

                  <!-- identifier pointing to the phenomenon definition -->
                  <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:1.0.30:Ozone">
                  <gml:metaDataProperty>

                     <!-- information about the offering to which this output is associated in
                     the SOS instance -->
                     <offering>
                        <id>Ozone</id>
                        <name>Ozone</name>
                     </offering>
                  </gml:metaDataProperty>

                  <!-- information about the unit of measurement in which the Ozone
                  concentration is delivered -->
                  <swe:uom code="ug/m3"/>
               </swe:Quantity>
            </sml:OutputList>
         </sml:outputs>
      </sml:System>
   </sml:member>

</sml:SensorML>
```