SensorNanny
SOS basic transactional server

Requirements

Table of Contents

1 Scope

The SOS server supports the following requests:

- getCapabilities
- describeSensor
- getObservation
- getObservationById
- getResult
- insertSensor
- deleteSensor
- insertObservation
- deleteObservation (not in official SOS protocol ?)
- insertResult
- deleteResult

The functions are implemented in a minimalistic way (only REST requests).

SWE formats (sensorML and O&M) to be submitted to the server (insertSensor, insertObservation, insertResult). The server validates and archives the records on a back-end storage facility (file system or no-SQL database).

The records can be deleted (deleteSensor, deleteObservation, deleteResult)

The records can be retrieved (describeSensor, getObservation, getObservationById).

When a request is not supported an explicit message is returned.

Important note (REST-FULL interface):
A persistent URL system is managed to resolve the sensors,observations and results records ids:
`http://sensornanny.ifremer.fr/record/<uuid>`
(in test `http://localhost:8080/sensornanny/record/<uuid>`).

The URL is resolved into the XML content of the record.

If the uuid is not found in the system:
`<error>record [uuid] is not available</error>`

To be done in V1: provide one URL with gives the list of available procedures and for each procedure the list of observations (O&M). See also the 52north SOS RESTFUL implementation.

For a version 0, some requests are not implemented:
getObservation
getResult
insertResult
deleteResult

2 Requests

2.1 GetCapabilities

Request

URL
http://<server-host>:<port>/sensorNanny/sos?service=SOS&version=2.0&**request=getCapabilities**

Returns an xml file stored locally.

To Be Done V1: add a function to provide the list of available procedures on server.

2.2 DescribeSensor

Request

URL
http://<server-host>:<port>/sensorNanny/sos?service=SOS&version=2.0&**request=describeSensor&procedure=<procedure id>&responseFormat=<format>**

responseFormat

application/json;subtype="http://www.opengis.net/om/2.0"
text/xml;subtype="http://www.opengis.net/om/2.0"

Procedure id is the persistent URL of the procedure (e.g. http://sensornanny.ifremer.fr/record/<uuid>)

Returns the sensorML record related to the procedure uuid in the format requested.

Errors
procedure id <procedure id> >is not available on server

responseformat <*responseFormat*> is not supported by the server. See getCapabilities for details.

2.3 GetObservation

Request

URL
http://<server-host>:<port>/sensorNanny/sos?service=SOS&version=2.0&**request=getObservation&procedure=<procedure id>&responseFormat=<format>**

Procedure id is the persistent URL of the procedure (e.g. http://sensornanny.ifremer.fr/record/<uuid>)

spatial and temporal criteria: To Be completed Only managed with no-SQL back-end.

Offering is the group of observation to which the requested observation belongs.

responseFormat

application/json;subtype="http://www.opengis.net/om/2.0"
text/xml;subtype="http://www.opengis.net/om/2.0"

Returns the sensorML record related to the procedure uuid in the format requested.

Errors

procedure id <procedure id> >is not available on server

responseformat <*responseFormat*> is not supported by the server. See getCapabilities for details.

2.4 getObservationById

Request

URL
http://<server-host>:<port>/sensorNanny/sos? service=SOS&version=2.0&**request=getObservationById&observation=<observation id>&responseFormat=<format>**

Observation id is among those available on the server (e.g. http://sensornanny.ifremer.fr/record/<uuid>)

responseFormat

application/json;subtype="http://www.opengis.net/om/2.0"
text/xml;subtype="http://www.opengis.net/om/2.0"

Returns the O&M records related to the observation uuid in the format requested.

Errors
observation id <observation id> >is not available on server

responseformat <*responseFormat*> *is not supported by the server. Supported formats are:*
...

2.5 InsertSensor

Request (only POST supported)

URL
http://<server-host>:<port>/sensorNanny/sos?service=SOS&version=2.0&request=insertSensor

(should work without any key value pair parameters (e.g. request=insertSensor).

QUERY STRING

```
<?xml version="1.0" encoding="UTF-8"?>
<swes:InsertSensor service="SOS" version="2.0.0"
   xmlns:swes="http://www.opengis.net/swes/2.0"
   xmlns:sos="http://www.opengis.net/sos/2.0"
   xmlns:swe="http://www.opengis.net/swe/2.0"
   xmlns:sml="http://www.opengis.net/sensorml/2.0"
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:gmd="http://www.isotc211.org/2005/gmd"
   xmlns:gco="http://www.isotc211.org/2005/gco"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/sos/2.0
http://schemas.opengis.net/sos/2.0/sosInsertSensor.xsd   http://www.opengis.net/swes/2.0
http://schemas.opengis.net/swes/2.0/swes.xsd">
   <swes:procedureDescriptionFormat>[format]</swes:procedureDescriptionFormat>
<swes:procedureDescription>
      [sensorML record]
</swes:procedureDescription>
<swes:observableProperty></swes:observableProperty>
</swes:InsertSensor>
```

Format is among:

application/json;subtype="http://www.opengis.net/om/2.0"
text/xml;subtype="http://www.opengis.net/om/2.0"

sensorML record is the sensor description in the format above (xml or json).

Actions:

- Validates the sensorML record
- Store (including convert to JSON or XML when necessary)
- Update the component list of the host system (information in 'attachedTo' tag).

Returns the validation status (3) and insertion status (4.1).

2.6 deleteSensor

Request

URL
http://<server-host>:<port>/sensorNanny/sos? service=SOS&version=2.0**&request=DeleteSensor&procedure=<procedure uuid>**

Actions:

- Delete the sensorML record having the requested procedure uuid

Returns status:

sensorML record <procedure uuid> has been successfully deleted.

Error:
When the requested procedure uuid is not available on local system:

sensorML record <procedure uuid> does not exist on server.

2.7 InsertObservation

Request (only POST supported)

URL
http://<server-host>:<port>/sensorNanny/sos?service=SOS&version=2.0&request=insertObservation

(should work without any key value pair parameters (e.g. request=insertObservation).

QUERY STRING
<?xml version="1.0" encoding="UTF-8"?> <sos:InsertObservation service="SOS" version="2.0.0" xmlns:sos="http://www.opengis.net/sos/2.0" xmlns:swes="http://www.opengis.net/swes/2.0" xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:sml="http://www.opengis.net/sensorML/1.0.1" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:om="http://www.opengis.net/om/2.0" xmlns:sams="http://www.opengis.net/samplingSpatial/2.0" xmlns:sf="http://www.opengis.net/sampling/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sos.xsd http://www.opengis.net/samplingSpatial/2.0

```
http://schemas.opengis.net/samplingSpatial/2.0/spatialSamplingFeature.xsd">
   <!-- multiple offerings are possible -->
   <sos:offering>[Offering]</sos:offering>

   <sos:observation>
     [O&M record]
   </sos:observation>
</sos:InsertObservation>
```

To Be done in V1: to enable upload of JSON records the format need to be a query criteria (XML or JSON as in insertSensor (see 2.5), BUT IT IS NOT AVAILABLE IN THE STANDARD.

Offering is the group of observation (dataset) to which the current observation is submitted (for example frenchResearchVessels). It is not managed here

O&M record is the observation description in XML ~~the format above (xml or json)~~.

Actions:

- Validates the O&M record
- Store (including convert to JSON or XML when necessary)

Returns the validation status (3) and insertion status (4.1).

2.8 deleteObservation

Request

URL
http://<server-host>:<port>/sensorNanny/sos? service=SOS&version=2.0**&request=DeleteObservation&observation=<observation uuid>**

Actions:

- Delete the sensorML record having the requested procedure uuid

Returns status:

sensorML record <procedure uuid> has been successfully deleted.

Error:
When the requested procedure uuid is not available on local system:

sensorML record <procedure uuid> does not exist on server.

2.9 insertResult

To Be completed

2.10 deleteResult

To Be completed

3 Validation

The validation steps are:

- validate with remote XSD
- validate with local schematron (if XSD is successful)

The validation ends with an OK status, otherwise it sends an error message:

XSD schema validation successful Schematron failed <schematron output>

Or

XSD schema validation failed <xsd validation output> Schematron not applied

Or

XSD schema validation successful Schematron validation successful

3.1 XSD validation

For sensorML records, the validation uses:

http://schemas.opengis.net/sensorML/2.0/sensorML.xsd

For O&M records, the validation uses:

http://schemas.opengis.net/om/2.0/observation.xsd

3.2 Schematron validation

For sensorML records, the validation uses:

http://www.ifremer.fr/isi/seadatanet/swe/sensorML/schematron/sensorml-sdn-core.sch

For O&M records, the validation uses:

http://www.ifremer.fr/isi/seadatanet/swe/om/schematron/om-sdn-core.sch

4 Back-end storage

4.1 Scope

The back-end storage manages the local repository where the records are persisted.

When new records are submitted to the back-end the result may be:

record <uuid> successfully imported

Or

record <uuid> successfully updated

Or:

error on record import: ….

4.2 Implementations

4.2.1 File system

The file system storage is organized as follow:

- one root directory
- every sensorML records
- when om are available for the a procedure (sensorML record), one subdirectory is created (named with the procedure id) and the O&M records are stored in it.

In each subdirectory the files are stored in XML and named after their UUID.

If records are submitted in JSON format, they are converted to XML.

```
/sensorNanny/data/
/sensorNanny/data/
/sensorNanny/data/0f088e5f-e0ad-4936-9024-7b5c9a552b0a.xml
/sensorNanny/data/18f16aca-4bb0-465b-9dcc-2861496bb99f.xml
/sensorNanny/data/18f16aca-4bb0-465b-9dcc-2861496bb99f/
/sensorNanny/data/18f16aca-4bb0-465b-9dcc-2861496bb99f/547df307-5ce8-431b-8a12-
8e8e51e23183.xml
/sensorNanny/data/18f16aca-4bb0-465b-9dcc-2861496bb99f/a9d0ac5e-656d-486d-822c-
f7fbb2d798f4.xml
/sensorNanny/data/18f16aca-4bb0-465b-9dcc-2861496bb99f/c07e0c74-c395-446a-b630-
4f57b8023a3b.xml
/sensorNanny/data/22113995-cd7a-4f39-a7d7-7e5c9f630d48.xml
...
```

4.2.2 No-SQL (e.g. couchDB)

To Be completed.

If records are submitted in XML format, they are converted to JSON.

5 Use Case Scenario

1. Submit network sensor (e.g. ifremer r/v): *insertSensor*
2. Submit sensor in network: *insertSensor*. Note “AttachedTo tag is used to update component list of the host system”.
3. Create offering : *configuration*
4. Submit observation (from one already inserted procedure) to offering: *insertObservation*

5. Browse sensor from network: *describeSensor*
6. Browse observation from sensor. *getObservationById*

6 ANNEX: JSON to XML revesible conversion

So that the XML to JSON translation is fully reversible, it has been chosen to convert XML to JSONArray objects.

It is easily implemented by using the JAVA API: http://www.json.org/java/index.html

JSON sensorML indented example:


```
[
  "sml:PhysicalSystem",
  {
    "xmlns:swe":"http://www.opengis.net/swe/2.0",
    "xmlns:gml":"http://www.opengis.net/gml/3.2",
    "gml:id":"top",
    "xmlns:xsi":"http://www.w3.org/2001/XMLSchema-instance",
    "xmlns:gco":"http://www.isotc211.org/2005/gco",
    "xmlns:swes":"http://www.opengis.net/swes/2.0",
    "xmlns:xlink":"http://www.w3.org/1999/xlink",
    "xmlns:gmd":"http://www.isotc211.org/2005/gmd",
    "xsi:schemaLocation":"http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sosInser
tSensor.xsd   http://www.opengis.net/swes/2.0 http://schemas.opengis.net/swes/2.0/swes.xsd",
    "xmlns:sml":"http://www.opengis.net/sensorml/2.0",
    "xmlns:sos":"http://www.opengis.net/sos/2.0"
  },
  [
    "gml:description",
    "PROVOR ARVOR  - 1000 dbar"
  ],
  [
    "gml:name",
    "PROVOR ARVOR  - 1000 dbar"
  ],
  [
    "sml:identification",
    [
      "sml:IdentifierList",
      [
        "sml:identifier",
        [
          "sml:Term",
          {
            "definition":"http://www.ifremer.fr/tematres/vocab/index.php?tema=66"
          },
          [
```

```
            "sml:label",
            "uuid"
          ],
          [
            "sml:codeSpace",
            {
              "xlink:href":"http://ubisi54.ifremer.fr/cgi-bin/sos.py?request=getCapabilities"
            }
          ],
          [
            "sml:value",
            "c07e0c74-c395-446a-b630-4f57b8023a3b"
          ]
        ]
      ],
    (...)
          [
            "sml:outputs",
            [
              "sml:OutputList",
              [
                "sml:output",
                {
                  "name":"PRES"
                },
                [
                  "swe:Quantity",
                  [
                    "swe:uom",
                    {
                      "code":"deciBar",
                      "xlink:href":"http://vocab.nerc.ac.uk/collection/P24/current/POWAREA/"
                    }
                  ]
                ]
              ]
            ]
          ],
          [
            "sml:method",
            [
              "sml:ProcessMethod",
              [
                "swe:extension",
                "where ser# = 3016 temperature coeffs: A0 =  -0.0000 A1 =   0.0003 A2 =  -0.0000 A3
=   0.0000"
              ],
              [
```

```
                    "swe:description",
                    "Temperature ITS-90 = 1/ { a0 + a1[lambda nu (n)] + a2 [lambda nu^2 (n)] + a3 [lamb
da nu^3 (n)]} - 273.15 (deg C)"
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
```