# SUDPLAN's Experiences with the OGC-Based Model Web Services for the Climate Change Usage Area

P. Kutschera, M. Bartha, D. Havlik

# SUDPLAN's experiences with the OGC-based model web services for the Climate Change usage area

P. Kutschera, M. Bartha and D. Havlik

Austrian Institute of Technology - AIT,
Donau-City-Straße 1, A-1220 Vienna, Austria
{peter.kutschera,mihai.bartha,denis.havlik}@ait.ac.at
http://www.ait.ac.at

**Abstract.** SUDPLAN is currently developing a technical solution for Model Web/Observation Web in the Climate Change usage area. Proposed solution is based on the Open Geospatial Consortium standards, and follows the ideas expressed in SANY Sensor Service Architecture (SensorSA). SUDPLAN also continued the development of the SANY software, resulting in native SOS, SPS and 2D coverage support in "Time Series Toolbox" framework for building sensor web applications. SUDPLAN re-uses much of the OGC SWE and SANY SensorSA functionality to: (1) configure and run the models; (2) provide the data (observations) required for model execution; (3) inform the user on model run progress and (4) access the model results. In this paper, we shall describe the SUDPLAN's experiences with implementing of the interoperable Model Web using OGC standards, and discuss the advantages of various services from the OGC SWE suite as compared to non-SWE alternatives in the Climate Change context.

**Keywords:** Environmental modelling; Open Geospatial Consortium; Sensor Web enablement; OGC SWE; Model Web; Observation Web; Time Series Toolbox; Sensor Service Architecture; SensorSA

## 1 Introduction

SUDPLAN is an EU FP7 project under the Information Communication Technology programme (ICT-2009-6.4), running 2010-2012. The project responds to the calls target "ICT for a better adaptation to climate change" which asks for solutions that combine advanced environmental modelling and visualization, in support to EU initiatives like The "Shared Environmental Information System" (SEIS) [22] and The "Single Information Space in Europe for the Environment" (SISE) [28].

The on-going implementation of the *Infrastructure for Spatial Information in Europe* directive (INSPIRE) [21], the transition from research to operations of the *Global Monitoring for Environment and Security* (GMES) initiative[23], the development of a *Shared Environmental Information System* (SEIS), and the combination of all three as a European contribution to the *Global Earth Observation System of Systems* (GEOSS) initiative [25], are profoundly changing the design of environmental applications. In order to allow re-using of investments across usage areas, organisations and applications, the researchers and application developers are required to provide access to "their" data and other functional building blocks (e.g. processing, visualization) through standardized web service interfaces.

*Sensor Web Enablement* suite of standards developed by *Open Geospatial Consortium* (OGC SWE) [20] already provides much of the required functionality for sensors and sensor observation archives; the SANY *Sensor Service Architecture* (SensorSA) [11] and the "Model Web" [24] envision the similar functionality for environmental models, and the newly coined "observation web" [26] extends these ideas to observations provided by humans.

## 2 Data, transport, and model runs in Service Oriented Architecture

Service-oriented architecture (SOA) is widely accepted as the paradigm of choice to loosely couple software components in distributed applications [29, 10].

Applied to numerical models, the SOA paradigm leads to idea of "Model Web", where model engines, as well as the resources required by these models are consequently encapsulated behind service interfaces and re-usable across a range of applications.

This section summarizes some of the key SOA requirenments. A more detailed discussion of this topic is given in ORCHESTRA RM-OA [10] and SANY fusion architecture documents [12]

1. Discovery of data and models With a notable exceptions of the RESTfull web services[33] and the LinkedData approach [19], the web services remain invisible for end users and web crawlers.
   Model Web resource discovery therefore requires special-purpose catalogs offering structured information on available resources. At the time of writing this paper, no fully functional generic solution for data and model discovery exists, but the future INSPIRE and GMES catalogues are expected to provide much of the required functionality. Comprehensive knowledge archive network (CKAN, http://ckan.net) search engine provides an interesting alternative to both Google and specialised catalogues for the Linked Data cloud.
   Data and service discovery is not on the SUDPLANs' research agenda, and therefore will not be further considered in this paper.
2. Formal definition of the model, required input data, parameters and output values (not shown in Fig. 1).
   SOA paradigm foresees re-using services and resources in applications they were not initially designed for. Consequently, each service needs provide formal self-description of its methods, as well as of the specific functionality

provided by the service. Model web services should also provide the estimates about input- and output- data sizes and expected runtimes.

3. Handling of the large data sets
   Model web services consume data from one or more sources on input side, process it and produce new data set(s) on the output side. Furthermore, the processing is controlled by parameters.
   Examples of input- and output- data relevant to SUDPLAN include climate-relevant sensor observations, pollution and rainfall patterns. Both the input data consumed by models, and their output can be large compared to the network throughput and the capacity of data services to deliver the required data. Consequently, the architecture must provide a way to replicate the data required for the model runs and to keep the local copy synchronised. An interesting discussion of the data replication and syncronization in OGC SWE networks has been given by Havlik et al. [27].
4. Model execution
   SOA services are typically idle unless explicitly triggered to perform some action. Model web services are often trigered by providing a set of parameters, but the model execution may be also performed separately from model configuration. Model parameters define the temporal and spatial constraints, as well as the initial conditions for the model run. In the model web context, the parameters may also allow the users to choose one of the available processing algorithms, decide which input data to use, or to schedule the delayed model execution.
5. Progress monitoring and control
   Some models are capable of calculating the results almost imediately. Other, including the SUDPLANs' downscalling services may require weeks to complete the task.
   This results in the need for monitoring the model progress and notifying the users of model status changes.
6. Download the results, which are also potential large datasets
   The type and size of the model result differes in a wide range. From just a few numbers to a timeseries of 3D grids in the scale of 100GB.
   So it makes no sense to return each result to the client. Instead it is stored on the server site and the client gets informed of when and where (parts of) the results can be downloaded.

## 3 Selection of standards to be used by SUDPLAN

All data need to be self describing [30] which means that there is information about the values like unit of measurement, description of phenomena (e.g. a URN of a ontology), precision and uncertainity and methods of measurement in the form of sensor descriptions. The same is true for models as we expect to get more models over time and they should be usable without prior knowledge of any details. There is a lot of standards for data transport and remote service

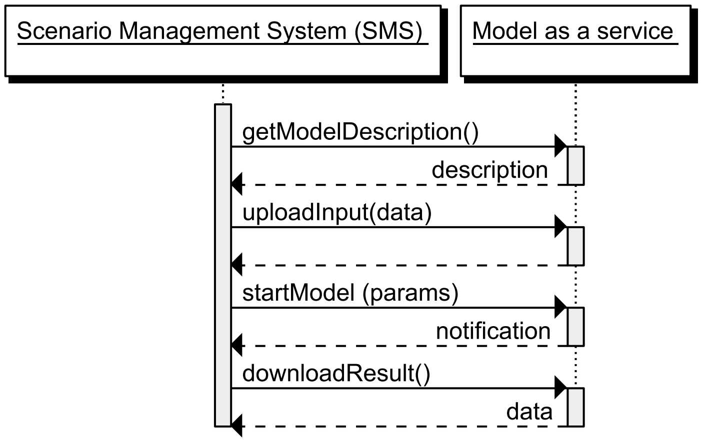


**Fig. 1.** The overall picture of a model invocation

invocation. SANY SensorSA[31] and Fusion Architecture[12] documents already foresee most of this.

Within SUDPLAN we concentrated to the OGC service interfaces. The main reasons where:

- The OGC service interfaces are an accepted standard in the GIS comunity, and cover a large number of use-cases relevant to environmental usage area
- All OGC specifications are freely available to everyone
- All OGC service specifications share common data encodings (O&M - Observation and Measurement[6]) and descriptions (GML - Geography Markup Language[2], SensorML - Sensor Model Language[8]).
- There is a lot of service interfaces to select from
- Implementations of libraries and some ready to use services are already available as open source

Now we will discuss which of the OGC interfaces are used in SUDPLAN and why we decided to use them.

### 3.1 Data transport: SOS, WCS, WFS, WMS?

Most of the data used in SUDPLAN fall into one of the following two categories:

- timeseries of scalar values like e.g. rain data from one measurement station, and
- timeseries of field data, e.g. mean temperature changes over whole europe for the next 100 years, one field every 10 years, one value in the field representing a 100x100 km sqare over Europe.

As discussed in the previous section, the associated meta-information such as units, spatial and temporal references hs to be provided together with the data.

Consequently, most of the SUDPLAN data can be thought of as timeseries of observations [31].

We needed to find a service interface that is generic enough to transport all out data but as specific as possible. Following OGC services were considered in SUDPLAN:

**WMS - Web Map Server** The main goal of the WMS[1] is to deliver maps to be visualized by a client. There is a lot of clients available ranging from browser-based interfaces to complete GIS systems.

The focus of the maps is the visualization, so the information is rendered and deliverd as images. In SUDPLAN the results from one model is often the input to the next model so images are not the data format of our choice. Also images are not the best data representation when comparing different climate or planning scenarios or when doing some more sophisticated visualizations.

Therefore the use of WMS in SUDPLAN is limited to background maps and quick overviews of the model results. The best candidate for this are the precalculated Climate Scenario data, which are used to show SUDPLAN users what climate changes to expect.

**WFS - Web Feature Service** A WFS[3] provides GML-encoded data with geographic reference.

This is a very powerfull service and many GIS systems can access a WFS to get data to render on a map.

WMS is often used to access shape files. While it would be possible to use WMS for accessing data in SUDPLAN, this standard is not optimised for handling timeseries of observations.

**WCS - Web Coverage Service** The WCS[5] is in many aspects similar to WMS.

Unlike WMS, the WCS return a grid of values suitable for further processing.

The usage of WCS as a way to access coverages was discussed within SUDPLAN. However, WCS is not optimised for timeseries of simple observations (unlike SOS). WCS remains an option for later extensions as at this early project state we tried to reduce complexity and implementation overhead.

**SOS - Sensor Observation Service** SOS[4] has been designed to provide access to sensor observations. Many sensors repeatedly observe the same phenomenon, so the result is typically a timeseries of observations, each associated to a time stamp. A model, or other processing entity, can be treated similarly to a sensor. Same is valid for the model input, output, parameters and identification information[12, **?**].

Although SOS appears to be the obvious service interface for data access in SUDPLAN, the encoding of continuous coverages has not been well-defined in

OGC SWE version 1.0. SOS relies on O&M information model, and the SUDPLAN approach for encoding continuous coverages, within limitations of the SOS 1.0 and O&M 1.0, is discussed in 3.3 section of this paper.

### 3.2 Model invocation: SPS, WPS?

Two of the OGC service specifications are suitable for initiation of remote program runs: Web Processing Service (WPS) and Sensor Planning Service (SPS).

**WPS - Web Processing Service** The WPS, initially named "Geoprocessing Service", was first introduced 2005. The WPS v1.0 (2007)[9] defines the web service interface and the mechanisms necessary to execute processes (analisys of georeferenced data) as well as to publish process dectiptions. The three WPS operations (GetCapabilities, DescribeProcess and Execute), providr enough flexibility to easily handle most processing use cases. Unfortunately, the WPS's flexibility of defining provider and domain specific input and output formats comes at the cost of the client which must support these formats.

**SPS - Sensor Planning Service** The more capable and complex SPS v1.0 was introduced in 2007[8] with the goal to describe sensor platforms and to allow the planning and scheduling of complex tasks in the context of earth observation satellites. While the initial goal of SPS seems somewhat distant from our goal of runing models subsequent research has shown that SPS can be sucesfully used as a general purpose interface for models and data fusion services.

In comparison to the WPS the SPS standard is more structured, and provides pre-defined mechanisms for client notification (status and result availability). The GetFeasibility operation in the SPS standard allows for client feedback, on the viability of a processing operation, before committing to execution. The SUDPLAN requirements on model description with formal specifications of parameters, input data and results, can be fulfilled by both WPS and SPS. However, the requirements on scheduling, cancelling, and monitoring the progress of the model runs can only be fulfilled by SPS. Further advantage of the SPS over WPS lies in the built-in notification handling.

#### Usage of OGC services for model invocation SUDPLAN

### 3.3 O&M encoding of timeseries of fields

As mentioned earlier, based on the size of the data, a distinction is made between parameters for a model run and input data. While the parameters are of simple type and small in size the input data for a model is complex in nature and quite common of several GB in size. Because the model needs a fast access to the input data this data has to be uploaded to the model site. The user of the model does this (sometimes over institutional boundaries) through an implementation of the

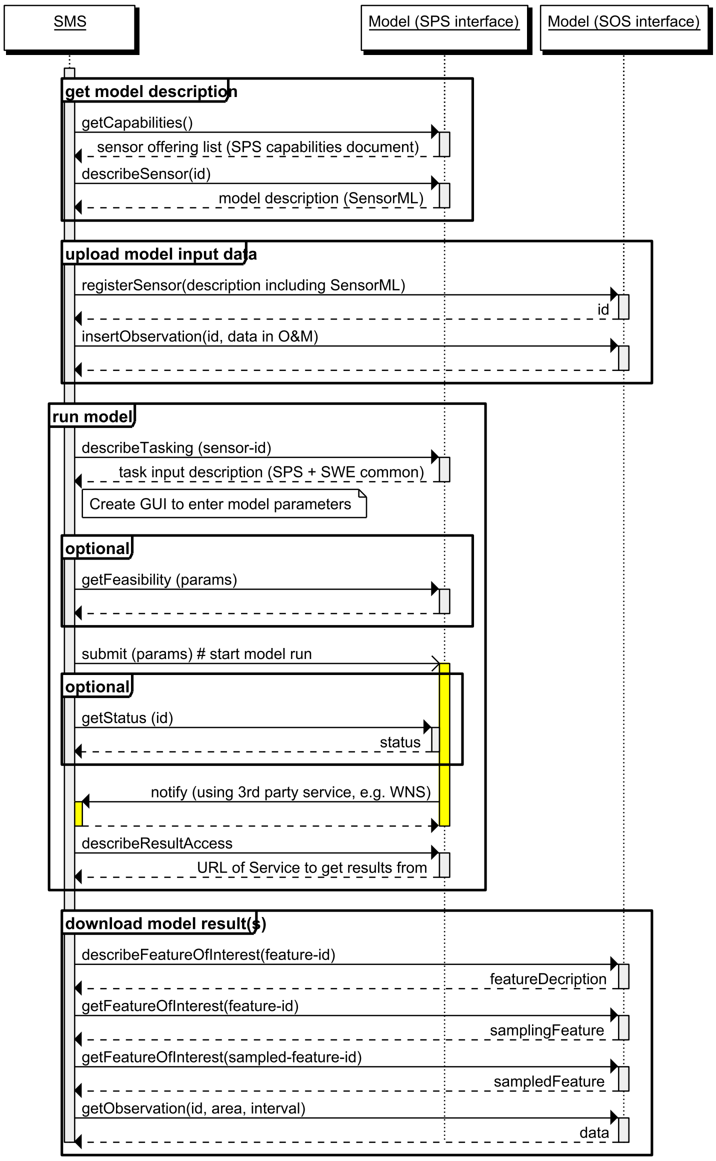


**Fig. 2.** Model invocation details showing OGC interface usage

SOS interface. This interface is also used to enumerate and retrieve the model results. Model result data transfered through the SOS interface is encoded using the O&M [6] information model.

The O&M encoding is straight forward for timeseries of scalar values and used by many SOS implementations (52North[16], OOSTethys[17]). The encoding of timeseries of coverages is not that well defined. Observations and Measurements - Sampling Features (O&M-SF)[7] specification provides means to encode discrete coverages and provides dedicated elements to do so, but it lacks dedicated elements necessary to describe continuous coverages.

SANY "Fusion and Modelling Architectural Design"[12] document describes two methods for encoding coverages in O&M. Both are using the O&M-SF sa:SamplingSurface element. While there is no dedicated element in O&M-SF v1.0 to specify a grid on which the sampling takes place, both methodes describe the sampling points and grid by embeding its description in sub-elements of the sa:SamplingSurface, which was not designed for this purpose.

In SUDPLAN a further method has been considered for describing continuous coverages.The optional SOS method DescribeFeatureOfInterest allows for retrieval of the feature type description (xml schema) for a given feature of interrest. This allows us to introduce a new namespace containing additional Sampling Feature types. The new SamplingGrid type inherits from the sa:SpatiallyExtensiveSamplingFeatureType defined in O&M-SF and contains a gml:RectifiedGrid element that describes the rectified grid on which the sampling takes place. The advantage of the mentioned inheritance is that the sampledFeature relation is retained. SUDPLAN SamplingGrid schema and one SamplingGrid instance example are shown below.

SamplingGrid schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     xmlns:gml="http://www.opengis.net/gml"
     xmlns:sa="http://www.opengis.net/sampling/1.0"
     xmlns:aitsa="http://www.ait.ac.at/sampling"
     targetNamespace="http://www.ait.ac.at/sampling"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/gml"
     schemaLocation="./gml4sos.xsd"/>
  <import namespace="http://www.opengis.net/sampling/1.0"
     schemaLocation="http://schemas.opengis.net/sampling/1.0.0/
samplingManifold.xsd"/>

  <element name="SamplingGrid" type="aitsa:SamplingGridType"
     substitutionGroup="gml:_Feature"/>

  <complexType name="SamplingGridType">
    <complexContent>
      <extension base="sa:SpatiallyExtensiveSamplingFeatureType">
        <sequence>
```

```
        <element ref="gml:RectifiedGrid"/>
      </sequence>
    </extension>
  </complexContent>
 </complexType>
</schema>
```

SamplingGrid instance example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SamplingGrid xmlns="http://www.ait.ac.at/sampling"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:sa="http://www.opengis.net/sampling/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gml
      http://schemas.opengis.net/gml/3.1.1/base/gml.xsd
      http://www.ait.ac.at/sampling
      ./SamplingGridSchema.xsd ">
  <gml:description>Sweden grid</gml:description>
  <gml:name>Swenden Grid</gml:name>
  <gml:location />
  <sa:sampledFeature xlink:href="urn:MyOrg:feature:sthlm_1"/>
  <gml:RectifiedGrid dimension="2">
    <gml:limits>
      <gml:GridEnvelope>
        <gml:low>0 0</gml:low>
        <gml:high>50 60</gml:high>
      </gml:GridEnvelope>
    </gml:limits>
    <gml:axisName>x</gml:axisName>
    <gml:axisName>y</gml:axisName>
    <gml:origin>
      <gml:Point srsName="urn:x-ogc:def:crs:EPSG:3021">
        <gml:pos>6546000.0 1580000.0</gml:pos>
      </gml:Point>
    </gml:origin>
    <gml:offsetVector srsName="urn:x-ogc:def:crs:EPSG:3021">
0 2e-005</gml:offsetVector>
    <gml:offsetVector srsName="urn:x-ogc:def:crs:EPSG:3021">
2e-005 0</gml:offsetVector>
  </gml:RectifiedGrid>
</SamplingGrid>
```

### 3.4 Using UncertML to describe statistical data

The descriptive model language Uncertainty Markup Language - UncertML[32] developed by the INTAMAP project can be used to encode the accuracy of the observation collection. SUDPLAN rainfall downscaling service generates a 2D table of the predicted precipitation values for the total seasonal accumulation (TOT), maximum

30-min intensity (MAX) and frequency of occurrence (FRQ). Because of the statistical characteristics of this data, UncertML could be used to encode it, similar to the way uncertainities in sensor data and model outputs were encoded in SANY.

This would mean treating this data as description of the timeseries of rainfall, not as an independent result. At the same time this data has the characteristics of a time series, meaning that it provides the above mentioned statistical information for every season. At this point no decission has been taken on wether in SUDPLAN this data should be encoded as observations (using O&M), or as observation uncertainities (using UncertML).

### 3.5 Using SensorML to describe models and required parameters

The SOS and SPS interfaces provide process descriptions encoded in in SensorML, through describeSensor operation. In SUDPLAN, the process is a models and can be described as Non-Physical (pure) process. The information provided in the form of a SensorML document can be quite extensive encompassing model inputs, parameters, outputs, the model algorithm itself and details of the implementation module. Currently, our use-cases require descriptions of constant model parameters necessary for the human interpretation of the model results. This includes: model identification, responsible party, input and output model. Our processing services act as clients to several Sensor Observation Services.

The following simplified SensorML document shows basic model identification as well the inputs and outputs of the rain downscaling model used in SUDPLAN. The model takes as input a SOS offering name containing the historical rain measurements as well as a future timestamp around which the downscaling results are generated. The model outputs a timeseries of coverages with the result model described in the Downscaled_rain named element.

```
<sml:SensorML xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
   xmlns:swe="http://www.opengis.net/swe/1.0.1"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink" version="1.0.1"
   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/sensorML.xsd ">
 <sml:member>
  <sml:System gml:id="SUDPLAN_A1B3">
   <gml:description>Simple rain downscaling model</gml:description>
   <sml:identification>
    <sml:IdentifierList>
     <sml:identifier name="UID">
      <sml:Term definition="urn:x-ogc:def:identifier:OGC:uuid">
       <sml:value>urn:x-ogc:object:model:SUDPLAN:prec:A1B3</sml:value>
      </sml:Term>
     </sml:identifier>
     <sml:identifier>
      <sml:Term definition="urn:x-ogc:def:identifier:OGC:shortName">
       <sml:value>SUDPLAN A1B3</sml:value>
      </sml:Term>
     </sml:identifier>
```

```
     </sml:IdentifierList>
    </sml:identification>
    <sml:inputs>
     <sml:InputList>
      <sml:input name="ObservationOfferingName">
       <swe:Text />
      </sml:input>
      <sml:input name="centerTime">
       <swe:Time />
      </sml:input>
     </sml:InputList>
    </sml:inputs>
    <sml:outputs>
     <sml:OutputList>
      <sml:output name="Downscaled_rain">
       <swe:DataArray>
        <swe:elementCount>
         <swe:Count />
        </swe:elementCount>
        <swe:elementType name="CoverageType">
         <swe:DataRecord>
          <swe:field name="Timestamp">
           <swe:Time definition="urn:ogc:data:time:iso8601"/>
          </swe:field>
          <swe:field name="Grid">
           <swe:DataArray>
            <swe:elementCount>
             <swe:Count />
            </swe:elementCount>
            <swe:elementType name="value">
             <swe:Quantity
               definition="urn:ogc:def:property:OGC:1.0:precipitation">
              <swe:uom code="mm"/>
             </swe:Quantity>
            </swe:elementType>
           </swe:DataArray>
          </swe:field>
         </swe:DataRecord>
        </swe:elementType>
       </swe:DataArray>
      </sml:output>
     </sml:OutputList>
    </sml:outputs>
   </sml:System>
  </sml:member>
</sml:SensorML>
```

# 4 Different model types used in SUDPLAN

This section provides three examples of the SUDPLAN Common Services[13] illustrating the bandwith of the model characteristics. We concentrate on:

1. Required input data type and size
2. Need of parameters,
3. Expected runtime,
4. Type and size of result

## 4.1 Climate Scenario data

Climate Scenario Common Service provides access to precalculated results of the climate change model. The result are timeseries of fields, typically about 51x51 values, one grid per 10 years over a time period of 140 years. Each field can contain some values (e.g. Temp, NO2). The entire dataset is about (51 * 51 * 13 decades * 7 different observed properties) 250000 values.

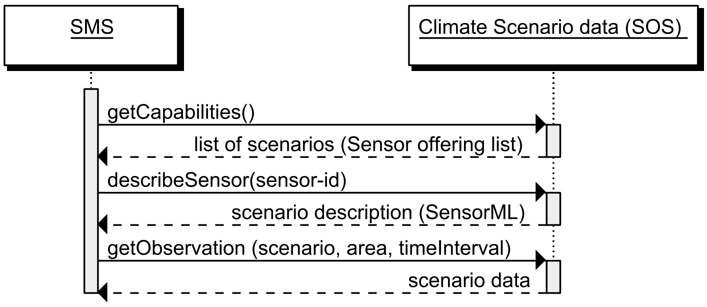


**Fig. 3.** Access Climate Scenario data

From the model invocation point of view the Climate Scenario Common Service is merely a data access service, as it does not provide any interface for model execution and control.

## 4.2 Rain timeseries downscaling

SUDPLAN rain timeseries Downscaling Service provides information on the local rainfall patterns for the next 100 years.

The input data is a historical timeseries of scalars, typically ten years of measured precipitation, one value every 5 minutes. This gives about one million values. Three parameters have to be provided by user for each model run: the name of the uploaded timeseries, the climate scenario to use, and the date for which a new timeseries shall be calculated[1].

[1] The location for which the future timeseries should be calculated is taken from the input timeseries.

The runtime of the model is short, just some minutes. The result is a timeseries with the same characteristics as the input timeseries, just moved into a possible future. Additional results are aggregated timeseries (e.g. one value every 30 minutes) and statistical information on the created data.

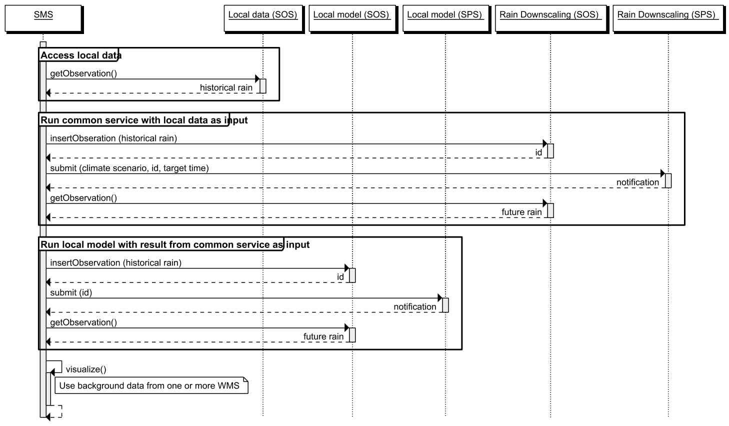


**Fig. 4.** Using downscaled rain timeseries as input to a local model

### 4.3 Air quality downscaling

Air quality Downscaling Service is the most important of the SUDPLAN Common Services. The model behind this service is able to perform local air quality downscaling based on local emission data.

The input is a field of expected emissions of greenhouse gases over an area of interest, typically a city. The parameters for this service are the climate scenario to use, the time interval and the area for which the air quality has to be calculated.

The runtime on a background supercomputer is expected to be around 10 days for a typical model run, and the result size is about 100GBytes. The model user has the possibility to download some result subsets, e.g. timeseries of grids, one value every 10 years or a timeseries for one location with higher temporal resolution. Available phenomena include temperature, precipitation, NO2 and more.

## 5 Software used

SUDPLANS aims to provide affordable downscaling services with standardised interfaces, and usable by every city in Europe. The main software components developed in SUDPLAN are:

### 5.1 Common Services

Most of the SUDPLAN Common Services already existed before the SUDPLAN project. These services are propriatery and executed on specialised supercomputers.

These services are accessed through standardized interfaces implemented by AIT using the open source TimeSeries Toolbox[18]. This implementation provides a wrapper around the models providing SOS interfaces for the transfer of data and SPS interfaces for executing and monitoring the models.

In addition, the WMS provides quick overviews of the model results. SUDPLAN uses the GeoServer[15] open source WMS implementation.

### 5.2 Scenario Management System

The Scenario Management System (SMS) is the main SUDPLAN user interface providing means to control models, retrieve and visualise results. SMS is based on cismet's CIDS system[14]. Both, CIDS as well an the SUDPLAN specific extensions are available under an open source licence.

### 5.3 Local models

SUDPLAN foresees two ways to extend the functionality of the system with additional local models: through SMS extensions, and through exposing of the local models through an SPS interface implementation analogously to the Common Services implementations described above.

Depending on the data type, a WFS (e.g. digital elevation model), a WMS (Aerial photos for visualisation) or a SOS (Sensor data) can be used. Open source implementations for all of these services are available from various sources.

## 6 Conclusion and outlook

This paper describes the current status of the SUDPLAN Model Web Implementation. SUDPLAN implementation bases of the OGC SWE 1.0 standards and on the SANY SensorSA idea of interfacing the models with SOS and SPS service interfaces. Our results confirm the suitability of the OGC SOS and OGC SPS services and related data encodings from the OGC SWE 1.0 standard suite for Climate Change related models and model results. SUDPLAN software related to interfacing of models and accessing the model results is avaliable as open source on ts-toolbox.ait.ac.at web site.

Some of the shortcomings mentioned in the paper have been adressed in SWE 2.0 development, in particular, the O&M 2.0 specification of the SamplingFeature. The SamplingGrid introduced in SUDPLAN provides an alternative method for describing the SamplingFeature (grid) of a continuous coverage. This method is superior to grid-encoding methods proposed in SANY, as it both retains the SampledFeature relation to the FeatureOfInterest and avoids the need of explicitly specifying all points of the grid. Furthermore, we believe that SamplingGrid is better aligned with O&M 2.0 specifications and provides a valid altenative to WCS encoding of continuous coverages.

SUDPLAN consortium is discussing the feasibility of using UncertML to describe statistic features of the downscaling results, and welcomes input on this topic.

## Acknowledgments

The work leading to the results presented in this paper has been partially funded through FP6 Integrated Project *Sensors Anywhere* (SANY; FP6 Contract number

033564; http://sany-ip.eu/) as well as through the FP7 *Sustainable Urban Development Planner for Climate Change Adaptation* project (SUDPLAN; EC grant agreement nr. 247708; http://sudplan.eu).

## References

1. Ogc web map service interface (01 2004), http://portal.opengeospatial.org/files/?artifact_id=4756
2. Opengis geography markup language (gml) implementation specification (02 2004), http://portal.opengeospatial.org/files/?artifact_id=4700
3. Web feature service implementation specification (05 2005), http://portal.opengeospatial.org/files/?artifact_id=39967
4. Opengis sensor observation service - implementation specification (02 2006), http://portal.opengeospatial.org/files/?artifact_id=26667
5. Web coverage service (wcs) implementation specification (10 2006), http://portal.opengeospatial.org/files/?artifact_id=18153
6. Opengis observations and measurements part 1 - observation schema (12 2007), http://portal.opengeospatial.org/files/?artifact_id=22466
7. Opengis observations and measurements part 2 - sampling features (12 2007), http://portal.opengeospatial.org/files/?artifact_id=22467
8. Opengis sensor model language (sensorml) implementation specification (07 2007), http://portal.opengeospatial.org/files/?artifact_id=21273
9. Opengis web processing service (06 2007), http://portal.opengeospatial.org/files/?artifact_id=24151
10. Reference model for the orchestra architecture (rm-oa) v2 (rev 2.1). Best Practice OGC 07-024, D3.2.3 of the European Integrated Project ORCHESTRA FP6-IST-511678 OGC 07-024, Open Geospatial Consortium (2007), http://portal.opengeospatial.org/files/?artifact_id=23286
11. Specification of the sensor service architecture v3 (rev. 3.1). Discussion Paper OGC 09-132r. Deliverable D2.3.4 of the European Integrated Project SANY FP6-IST-033564 OGC 09-132r, Open Geospatial Consortium (2009), http://portal.opengeospatial.org/files/?artifact_id=35888
12. Sany fusion and modelling architecture. Discussion Paper OGC 10-001, Deliverable D3.3.2.3 of the European Integrated Project SANY FP6-IST-033564 OGC 10-001, Open Geospatial Consortium (2010), http://portal.opengeospatial.org/files/?artifact_id=37139
13. Common services (03 2011), http://www.smhi.se/sudplan/About-SUDPLAN/About-SUDPLAN/common-services-1.10161
14. Geoinformationsplattform (03 2011), http://www.cismet.de/en/cidsfacts.pdf
15. Geoserver web map service documentation (03 2011), http://docs.geoserver.org/stable/en/user/services/wms/index.html
16. Sensor observation service (03 2011), http://52north.org/communities/sensorweb/sos/index.html
17. Sensor observation service (03 2011), http://www.oostethys.org
18. The time series toolbox overview (03 2011), http://ts-toolbox.ait.ac.at/TS-Toolbox/pdf/TS-Toolbox_Summary.pdf
19. Berners-Lee, T.: Linked data. International Journal on Semantic Web and Information Systems 4 (2006), http://www.w3.org/DesignIssues/LinkedData.html

20. Botts, M., Percivall, G., Reed, C., Davidson, J.: Ogc sensor web enablement: Overview and high level architecture (07 2007), http://portal.opengeospatial.org/files/?artifact_id=25562
21. EC: Directive 2007/2/ec of the european parliament and of the council of 14 march 2007 establishing an infrastructure for spatial information in the european community (inspire). EU Directive 2007/2/EC, European Community (03 2007), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF
22. EC: Communication from the commission to the council, the european parliament, the european economic and social committee and the committee of the regions - towards a shared environmental information system (seis) SEC(2008) 111SEC(2008) 112 . Tech. Rep. COM(2008) 46 final, European Commision (02 2008), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52008DC0046:EN:NOT
23. EP, Council: Regulation (eu) no 911/2010 of the european parliament and of the council of 22 september 2010 on the european earth monitoring programme (gmes) and its initial operations (2011 to 2013). Regulation 911/2010, European Parliament and Council (09 2010), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:276:0001:0010:EN:PDF
24. Geller, G.N., Melton, F.: Looking forward: Applying an ecological model web to assess impacts of climate change. Biodiversity, 9(3&4). 9(3 and 4), 79–83 (2008)
25. GEO: Geo 2009-2011 work plan. Tech. rep., Group on Earth Observations (12 2010), http://www.earthobservations.org/documents/work\%20plan/geo_wp0911_rev3_101208.pdf
26. Havlik, D., Schade, S., Sabeur, Z.A., Mazzetti, P., Watson, K., Berre, A.J., Mon, J.L.: From sensor to observation web with environmental enablers in the future internet. Sensors 11(3) (2011)
27. Havlik, D., Schimak, G., Bleier, T.: Cascading and replicating the ogc sensor observation service. In: Anderssen, R.S., Braddock, R.D., Newham, L.T.H. (eds.) Proceedings of the 18th World IMACS / MODSIM Congress. pp. 966–972. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, Cairns, Australia (07 2009), http://www.mssanz.org.au/modsim09/C4/havlik.pdf
28. Hebek, J., Pillmann, W.: Shared environmental information system and single information space in europe for the environment: Antipodes or associates? In: Hebek, J., Hradec, J., Pelikn, E., Mrovsk, O., Pillmann, W., I.Holoubek, Bandholtz, T. (eds.) Proceedings of the European conference of the Czech Presidency of the Council of the European Union, TOWARDS eENVIRONMENT. pp. 447–458. Masaryk University, Prague, (03 2009), http://www.epractice.eu/files/SEIS\%20and\%20SISE\%20for\%20the\%20Environment_Antipodes\%20or\%20Associates.pdf
29. ISO: Iso 19119 geographic information – services (2005)
30. Klopfer, M., Kanellopoulos, I. (eds.): orchestra - an open source architecture for risk management. the ORCHESTRA Consortium (2008), http://www.eu-orchestra.org/, ©2008 by the ORCHESTRA Consortium
31. Klopfer, M., Simonis, I. (eds.): SANY - an open service architecture for sensor networks. SANY Consortium (2009), http://www.sany-ip.eu/publications/3317
32. Matthew Williams, Dan Cornford, L.B.E.P.: Uncertainty markup language (uncertml) (2007), http://www.intamap.org/uncertml/uncertml.php
33. Rodriguez, A.: Restful web services: The basics. developerWorks (2008), https://www.ibm.com/developerworks/webservices/library/ws-restful/