Sensor Web Enablement based Model Web Implementation for Climate Change Applications

Mihai Bartha, Peter Kutschera, Denis Havlik

Abstract

Interoperable Web-enabled environmental models are essential part of the service oriented environmental applications. Concepts, architectures, standards and software allowing seamless integration of models in distributed web-enabled applications (Model Web) are therefore a logical extension of the quest for interoperable service infrastructures allowing visualization and data access that is currently implemented within the scope of Infrastructure for Spatial Information in Europe directive (INSPIRE), and necessary for further development of the Global Monitoring for Environment and Security (GMES), Shared Environmental Information System (SEIS), and the Global Earth Observation System of Systems (GEOSS) initiatives.

SANY Sensor Service Architecture (SensorSA) proposed the Sensor Web Enablement (SWE) suite of standards developed by Open Geospatial Consortium (OGC) as a base for web-enabling the "sensor-like models". Sensor-like in this context means that the output data model contains observations, that is a set of values with (at least) the associated units, spatial and temporal context. This is true for a great majority of environmental models, and for many other sources of information including e.g. cadastres and the Volunteered Geographic Information (VGI). This versatility of SWE, as well as its focus on processes rather than on the physical sensors is often confusing for end users. We therefore recommend use of "observation web" as a technology-neutral synonymous for sensor web in all discussions with end users and decision makers, and SWE-based Model Web for web-enabled models with SWE service interfaces and data models.

Our experiences in SANY and, more recently, in SUDPLAN research project show that SWE indeed provides most of the Model Web functionality for environmental and climate change applications. The Sensor Observation Service (SOS) allows us to access both the underlying data required for the model run, as well as to expose the model results; the Sensor Planning Service to configure, schedule and control the model runs; and the Sensor Modelling Language (SensorML) and Observations & Measurements (O&M) modelling languages to describe the process and the result set respectively.

This paper presents our experiences with SWE-based Model Web in SUDPLAN, and discusses the already achieved results, as well as the planned model web developments. Special attention shall be given to (1) process description in both SOS and SPS context; (2) various possibilities for presenting the time series of 2D coverages in O&M; (3) model parameterization, execution and monitoring; (4) input-, output- and processing uncertainties.

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,

1 Austrian Institute of Technology - AIT, Donau-City-Straße 1, A-1220 Vienna, Austria
email: Mihai.Bartha@ait.ac.at, Peter.Kutschera@ait.ac.at, Denis.Havlik@ait.ac.at, Internet: http://ait.ac.at
in support to EU initiatives like The "Shared Environmental Information System" (SEIS) (EC 2008) and The "Single Information Space in Europe for the Environment" (SISE) (Hebek/Pillmann 2009). The on-going implementation of the Infrastructure for Spatial Information in Europe directive (INSPIRE) (EC 2007), the transition from research to operations of the Global Monitoring for Environment and Security (GMES) initiative (EP 2010), 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 (GEO 2010), are profoundly changing the design of environmental applications. In order to allow re-using of investments across usage areas, organizations 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) (Botts/Percivall/Reed/Davidson 2007) already provides much of the required functionality for sensors and sensor observation archives; the SANY Sensor Service Architecture (SensorSA) (SANY 2009) and the "Model Web" (Geller/Melton 2008) envision the similar functionality for environmental models, and the newly coined "observation web" (Havlik/Schimak/Bleier 2009) 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 (ISO 2005, ORCHESTRA 2007). 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 requirements. A more detailed discussion of this topic is given in ORCHESTRA RM-OA (ORCHESTRA, 2007) and SANY fusion architecture documents (SANY 2010).

1. Discovery of data and models. With a notable exceptions of the RESTfull web services (Rodriguez 2008) and the LinkedData approach (Berners-Lee 2006), the web services remain invisible for end users and web crawlers. Model Web resource discovery therefore requires special-purpose catalogues 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 synchronized. An interesting discussion of the data replication and synchronization in OGC SWE networks has been given by (Havlik/Schimak/Bleier 2009).

4. Model execution. SOA services are typically idle unless explicitly triggered to perform some action. Model web services are often triggered 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 immediately. Other, including the SUDPLANs’ downscaling 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 differs 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.

![Figure 1 The overall picture of a model invocation.](image)

3. **SELECTION OF STANDARDS TO BE USED BY SUDPLAN**

All data need to be self describing (Klopfer/Kanellopoulos 2008) which means that there is information about the values like unit of measurement, description of phenomena (e.g. a URN of an ontology), precision and uncertainty 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 invocation. SANY Sensor-SA(Klopfer/Simonis 2009) and Fusion Architecture (SANY 2010) 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 community, 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 (OGC 2007a) and descriptions (GML - Geography Markup Language (OGC 2004b), SensorML - Sensor Model Language (OGC 2007c)).
- 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

The following section shows which of the OGC interfaces are used in SUDPLAN and the reasons of using 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 square over Europe.

As discussed in the previous section, the associated meta-information such as units, spatial and temporal references has to be provided together with the data. Consequently, most of the SUDPLAN data can be thought of as timeseries of observations (Klopfer/Simonis 2009).

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 Service.** The main goal of the WMS (OGC 2004a) 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 delivered 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 procomputed Climate Scenario data, which are used to show SUDPLAN users what climate changes to expect.

**WFS - Web Feature Service.** A WFS (OGC 2005) provides GML-encoded data with geographic reference. This is a very powerful 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 optimized for handling timeseries of observations.

**WCS - Web Coverage Service.** The WCS(OGC 2006b) 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 optimized 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 (OGC 2006a) 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 (SANY 2010, Klopfer/Simonis 2009). 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 in 2005. The WPS v1.0 (OGC 2007d) defines the web service interface and the mechanisms necessary to execute processes (analysys of georeferenced data) as well as to publish process descriptions. The three WPS operations (GetCapabilities, DescribeProcess and Execute), provide 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 (OGC 2007c) 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 running models subsequent research has shown that SPS can be successfully 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.

### 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 SOS interface. This interface is also used to enumerate and retrieve the model results. Model result data transferred through the SOS interface is encoded using the O&M (OGC 2007a) information model.
The O&M encoding is straightforward for timeseries of scalar values and used by many SOS implementations (52North 2011, OOSTETHYS 2011). The O&M encoding for timeseries of coverages is not that well defined. Observations and Measurements - Sampling Features (O&M-SF) (OGC 2007b) 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" (SANY 2010) 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 methods describe the sampling points and grid by embedding 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 sos:DescribeFeatureOfInterest allows for retrieval of the feature type description (XML schema) for a given feature of interest. This allows for the introduction of a new namespace containing additional SamplingFeature 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:

```xml
<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:
One of the SUDPLAN requirements concerns the ability to retrieve the timeseries of values for one region or one point on the SOS offered gridded data. This introduces several problems with the implementation of the SOS interface.

When retrieving the data for a region of the grid one needs to specify a spatial filter. For our case a two dimensional bounding box (BBox) is submitted as part of the getObservation request to the SOS server. As described above, upon retrieval of the feature of interest for the observation offering, the description of the SamplingGrid is returned. One problem is that while the SamplingGrid describes the extents of the whole grid for which data is available, the SamplingFeature of the spatially filtered getObservation result will not be the same. In this regard the featureOfInterest for the offering cannot be referenced within the result, instead the SamplingFeature corresponding to the spatialFilter has to be contained inline.

Second there is no simple way of introducing a point spatial filter (necessary for the getObservation request) that contains the grid coordinate for which the data is requested. Such an element is not defined in the SOS specification and extending the specification with such an element would result in interoperability problems. One possible solution for this problem is using the bounding box spatial filter (BBox) with an
area of zero (same value for the lower-left and upper-right). The straightforward result model would be that of a simple timeseries (as opposed to gridded timeseries result model) which conflicts with the SamplingGrid feature of interest element of the corresponding offering. The chosen solution here was to describe the featureOfInterest (containing the point) in the result. One further solution is encoding in the result a SamplingGrid with only one element (one times one) and its extents dependent on the use case (area for which the value is valid). This has the advantage of maintaining the relation to the featureOfInterest from the offering and having a unified way of providing gridded observations.

3.4 Using UncertML to describe statistical data

The descriptive model language Uncertainty Markup Language - UncertML (Williams/Cornford 2007) 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 uncertainties 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. This specific data can be encoded as observations (using O&M) or as uncertainties (using UncertML) of the observation. In SUDPLAN the decision has been taken to use UncertML express the relation between the timeseries and the resulting statistical data from the model run.

Example StatisticsArray:

```
<un:StatisticsArray xmlns="http://www.uncertml.org"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.uncertml.org/1.0.1/">
  <un:elementType>
    <un:StatisticsRecord>
      <un:field name="Begin">
        <swe:Time definition="urn:ogc:data:time:iso8601"/>
      </un:field>
      <un:field name="End">
        <swe:Time definition="urn:ogc:data:time:iso8601"/>
      </un:field>
      <un:field>
        <un:Statistic definition="http://dictionary.uncertml.org/statistics/total"/>
      </un:field>
      <un:field>
        <un:Statistic definition="http://dictionary.uncertml.org/statistics/maximum"/>
      </un:field>
      <un:field>
        <un:Statistic definition="http://dictionary.uncertml.org/statistics/frequency"/>
      </un:field>
    </un:StatisticsRecord>
  </un:elementType>
</un:StatisticsArray>
```
Using SensorML to describe models and required parameters

The SOS and SPS interfaces provide process descriptions encoded 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 retrieving input data and storing model results.

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.

```xml
<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: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:System>
    </sml:member>
</sml:SensorML>
```
<sml:IdentifierList>
</sml:IdentifierList>
<sml:identification>
</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 (SUDPLAN 2011) illustrating the bandwidth 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 precomputed 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.

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.²

² 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.

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 proprietary and executed on specialized supercomputers. These services are accessed through standardized interfaces implemented by AIT using the open source TimeSeries Toolbox (AIT 2011). 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 (Geoserver 2011) 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 (CISMET 2011). Both, CIDS as well as 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 available as open source on ts-toolbox.ait.ac.at web site.

Some of the shortcomings mentioned in the paper have been addressed 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 alternative 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.
7. Acknowledgements

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).

8. References


ISO (2005). Iso 19119 geographic information – services


