Geospatial Standards for
Web-enabled Environmental Models[*]

Patrick Maué[1], Christoph Stasch[1], George Athanasopoulos[2], Lydia E. Gerharz[1]

[1] Institute for Geoinformatics (ifgi), University of Münster, Weseler Str. 253, 48151, {firstname.surname}@uni-muenster.de

[2] Dept. of Informatics & Telecommunications, National & Kapodistrian University of Athens (NKUA), Panepistimiopolis, Ilisia 157 84, Greece, gathanas@di.uoa.gr

Abstract

Serving geographic information via standardized Web services has been widely accepted as a useful approach. Web-enabled environmental models simulating real-world phenomena are, however, rare. The models predict observations traditionally served by geospatial Web services compliant to well-defined standards. Using standardized Web services could support decoupling of models, comparison of similar models, and the automatic integration into existing geospatial workflows. Modeling experts face several open issues when migrating existing environmental computer models to the Web. The selection of the Web service interface depends on the input parameters required for the successful execution of the computer model. Losing control over the execution of the models, and consequently also the confidence in model results, can be addressed to a certain extent by using translucent and standardized workflow languages. Mechanisms and open problems for the implementation of geospatial Web service compositions are discussed. Two scenarios about oil spills and the exposure to air pollution illustrate the impact of unconfigured model parameters for standard-compliant spatial data clients.

Keywords: Model Web, Environmental Models, Spatial Data Infrastructures, Environmental Observations, Web Service Standards

---

1. INTRODUCTION

Environmental models facilitate the understanding of changes in the environment, for example, by forecasting the next day's weather, predicting the distribution of forest fires, or rendering soil maps from core samples. They support decision makers in understanding the impact of real world events and processes in geographic space, and help in evaluating appropriate response measures. Different variations of environmental models exist; simulating environmental systems to study the impact of change of certain parameters is one example. In this research, we limit the definition of environmental models to software implementing mathematical models representing real world processes. These models vary regarding their complexity. Simple mathematical models might interpolate missing values from known sensor observations, while complex models make use of current or historical sensor data to predict future conditions. Computer models are implementations of mathematical models for simulating (and replicating) real world processes. The more complex the computer model, the more parameters and data inputs are typically required for its execution. For example, a hydrodynamic model may only take the river's discharge as input to predict downstream discharge, whereas more sophisticated solutions consider the terrain and the land cover of adjacent areas (Kurzbach et al, 2009).

The ongoing implementation of INSPIRE (DIRECTIVE 2007/2/EC) and initiatives like GMES ("Global Monitoring for Environment and Security") and SEIS ("Shared Environmental Information System") call for a migration of environmental models to reusable Spatial Data Services (SDS). This will eventually lead to a large supply of environmental models in the Web. From this set of models, domain experts can select the most appropriate implementations in the context of specific scenarios. Environmental models could then be replaced if certain characteristics such as the result's accuracy do not match the application's requirements. Reaching this objective relies on environmental models implemented as SDS with well-defined access methods and standardized formats for the consumed and produced data. These standards specify a common encoding and enable support in either local Desktop GIS or large-scale SDIs ("Spatial Data Infrastructures"). The idea of the Model Web (Geller and Melton, 2008) takes this view even further. It envisions interoperable computer models, embedded in a multidisciplinary network of models, data sources, processes, and sensors. It is a proposal for dynamic infrastructures for environmental computer models serving researchers, managers, policy makers, and the general public.

Within GEOSS ("Global Earth Observation System of Systems"), the Model Web is defined as one of several approaches addressing current shortcomings for predicting the impact of, for example, the changing environment. GEOSS was established as international collaboration effort for enabling access to earth observations across national and semantic boundaries. The goal is to enhance

interoperability of existing models and to make their outputs more accessible. Similar to the conceptual architecture of SDIs, GEOSS implementations focus on modularity and interoperability to achieve "integration-ready" components (Christian, 2005). Despite the existing effort, domain experts are reluctant to move environmental models into the Web. Open issues such as capturing model semantics, communicating uncertainty of the results, and ensuring efficient execution and workflow modelling are (and have been) subject of the research projects ENVISION[1], UncertWeb[2], GDI-Grid[3], SWING[4], and SODIUM[5]. Concerns about trust (the risk of involuntarily sharing data and algorithms) and control (fear of losing control over model definition, execution, and calibration) are only partly discussed in this paper, but have to be considered for Web-enabling environmental models.

In this paper we recommend the use of standardized and well-established service interfaces to deploy and invoke environmental models on the Web. We introduce the distinction between pre-configured/non-interactive models and unconfigured/interactive models. This classification derives from the discussion about user expectations (i.e. "what do we expect as result from an environmental model?") and technical expectations (i.e. "how does a client request and visualize model results?"). This distinction is also exemplified by the two pilot cases which are presented in this paper. In addition, we discuss how existing environmental models running as stand-alone applications are decoupled into individual Web services representing either certain processing parts of the models or their input data. Common process modeling languages can then be used to couple these Web services into service compositions. These compositions support the flexible combination and reuse of the processing parts. In the remainder we refer to environmental models implemented as service compositions as *Environmental Process Models* (EPM). Granularity has to be considered as an issue for the decomposition of local computer models into Web services. Existing environmental models could either be exposed as one Web service or be split up into several constituent Web services representing the individual steps. Even though this aspect is of importance for the modeling expert, it is irrelevant for the presented architecture. This approach supports both, fine and coarse-grained compositions. The EPM can comprise one or many Web services coupled through a process modeling language. The resulting composition is exposed as a Web service compliant to the standards set by the Open Geospatial Consortium (OGC). The standards ensure the seamless integration of the models into existing geospatial workflows. The introduction of the architecture for this

---

[1]   European research project (FP7): http://www.envision-project.eu
[2]   European research project (FP7):   http://www.uncertweb.org/
[3]   German research project (BMBF):   http://purl.org/ifgi/projects/gdi-grid
[4]   European research project (FP6):   http://purl.org/ifgi/projects/swing
[5]   European research project (FP6):   http://www.atc.gr/sodium

approach and the discussion of the reasons for selecting the appropriate OGC standards should be considered as the main contribution of this paper. The paper also lists some of the unique aspects of geospatial workflows in general For example, geospatial applications are traditionally dealing with large amounts of data. This complicates concerns such as scalability and performance of the execution. An example implementation further illustrates that existing and well-established standards can be simply applied to model and invoke geospatial workflows.

In the following Section 2, the relevant standards and specifications from the OGC are introduced. In Section 3, we discuss user expectations for environmental models, the notion of workflows, the distinction between pre- and unconfigured models, and suggest appropriate service interfaces. Section 4 lists two scenarios to further illustrate this distinction and to highlight current research on uncertainty and semantics for geospatial workflows. The efficient execution of model workflows is explained in detail in section 5, followed by an example illustrating how the scenario could be implemented. In section 6 the presented approach is evaluated in relation to results of past and ongoing research projects. The conclusion in section 7 provides an outlook into future research, and summarizes the main findings of the paper.

2. STANDARDS FOR ENVIRONMENTAL MODELS

Spatial Data Services (SDSs) manage the access to geographic information. They are embedded in SDIs, which build the foundation for standardized and interoperable exchange of spatial data. Publishing EPMs as SDS has been recognized as useful (Mineter et al, 2003, Granell et al, 2010) to improve acceptance of model results and collaboration between environmental experts. Embedding EPMs in SDIs relies on interoperable components. This requires standards which are presented in this section.

2.1.   Standards for Spatial Data Services

EPMs usually result in spatial and/or temporal data, e.g. field-based predictions or time series. Elements of spatio-temporal data can be modeled as vector-based features, raster-based coverages, or sensor observations. Depending on the data model, commonly used and often standardized encodings exist. Typical formats for vector-based features are the OGC Geography Markup Language (GML) (Vretanos, 2005), or the ESRI Shapefile. GeoTIFF or NetCDF are popular choices for raster-based data. The OGC standard Observations & Measurements (O&M) defines an encoding for sensor observations delivered as time series (Cox, 2007). Sensor metadata is described in SensorML, the Sensor Model Language (Botts and Robin, 2007).

Specialized Web service interfaces have been specified by the OGC for accessing the different forms of spatial data. The OGC Web Feature Service (WFS) defines methods to retrieve vector-based geographic features encoded in GML. The Web Coverage Service (WCS) specifies access to raster-based data (Whiteside and Evans, 2007). However, temporal filtering of the retrieved environmental observations is not sufficiently supported by the WCS and WFS standards. The Sensor Observation Service (SOS) provides temporal filtering for observation time-series encoded in O&M. The benefits of standards have also been recognized for the processing of spatial data. Geoprocessing methods such as transforming or merging of geospatial data are managed by the Web Processing Service (WPS). The WPS interface provides operations to retrieve detailed information about the available processes, their parameters and their execution (Schut, 2007). All OGC Web service interfaces also include methods to retrieve metadata required for evaluating the usefulness of the offered spatial data or processes for the client's application. Common encodings and access methods enable generic spatial data clients to directly reuse SDS for common tasks such as geospatial analysis and decision-making. The EPM Web service interface and the encodings of the model results should comply with the introduced standards to enable the desired integration.

2.2.   Standards for Web Service Compositions

Decomposing the complex legacy models into Web services enables separation of input data and model algorithms. These Web services can then be reused in other scenarios (e.g. the model algorithm can be applied for another region using different data as input). The ISO 19119 standard defines coupling of (OGC) services as *service chaining*. Service workflows and service compositions are also commonly used and are considered equivalent for this research[6]. By far the most widely-used standard for the encoding of Web service process models is the Business Process Execution Language (BPEL) (Diane and Evdemon, 2007). BPEL is a declarative language that supports the description of either abstract or executable service processes. It provides high-level constructs (inherited from workflow languages) which enable the design of service workflows by domain experts. A wide range of tools exists for both, the composition of existing Web services into more complex workflows and the subsequent execution with a run-time engine.

BPEL provides the necessary means to realize process models as computer-executable models accessible over the Web. The idea of coupling existing SDS to sophisticated service compositions is still not more than a vision. Semantic conflicts, insufficient means to describe spatial data quality, over-complicated composition tools, and execution performance significantly impair widespread

---

[6]   Their slight difference is not relevant for the work introduced in this paper.

acceptance. These problems are relevant, but are not unique to spatial data. For the presented work, we assume they will be eventually solved. The authors have addressed some of these topics already in their recent work (Tsalgatidou et al 2008, Maué et al 2009, Athanasopoulos et al 2010).

3. BRINGING MODELS INTO THE WEB

In the following discussion we assume that environmental models predict or interpolate environmental observations. We explain which OGC standards should be selected for environmental models and how to use BPEL to implement these models. The distinction between pre- and unconfigured models is crucial for the workflow invocation and accordingly for the presented architecture.

3.1. Environmental Process Models as OGC Services

EPMs simulate real world phenomena occurring in geographic space. Exposing EPMs as Web services depends on the definition of its interface. Using an OGC standard as interface (as long as the implementations comply with the specifications and match in versions) enables:

Coupling models: One of OGC's self-claimed goals is to enable geoprocessing technologies to "plug and play". A streamflow model (for predicting the amounts of water transported by a river) might rely on real-time information about the river's water level. It expects a collection of sensor observations from river gauges. This data may come as well from a water runoff model predicting water levels by looking at real-time precipitation data in the river's catchment area. Standards ideally allow for simply chaining various models to create more sophisticated models (which, in the end, are needed to approximate the complexity of reality) or to overcome data sparseness. Standard interfaces do not only support the coupling and integration of environmental models as such, but also the replacement of the individual modules with equivalent services. But this implies, besides syntactic and structural interoperability, also a sound understanding of service semantics.

Comparing models: Environmental models have an impact on decisions of public authorities. The predictions of UK's National Weather service about the spread of the ash cloud caused by the 2010 eruption of Eyjafjallajökull volcano lead air traffic controls to close down large parts of the European airspace. This decision was followed by critical claims of over-reaction and doubts about confidence in the model results. Making not only the results but also the model algorithms and the data sources available as Web services supports verification. Furthermore, the availability of different models for predicting similar phenomena supports identification of significant differences (and accordingly uncertainty of the predictions).

Integrating models: Using standardized interfaces supports integration into common spatial data applications like GIS software. This allows easy integration of models into existing scientific (or enterprise) workflows, supporting reuse and acceptance within the modelling community.

Unfortunately, these points do represent not the reality, but the potential of standards. A plethora of different versions across implementations, different and sometimes conflicting understandings of how to implement specifications, and the trend of making standards too complex, impair the envisioned seamless integration. Nonetheless, we believe that OGC standards can help to ensure a certain level of interoperability. The listed benefits explain why common OGC standards should be used for environmental models. In the next section, we propose one particular OGC standard for encoding the output of EPMs.

3.2. Environmental Process Models as Virtual Sensors

Physical sensors serve environmental observations such as measurements for precipitation or the concentration of pollutants in the air. The EPMs considered here provide approximations of these observations, e.g. by interpolating precipitation samples to assess weather conditions at not observed locations. Sgroi et al (2005) define virtual sensors as "objects that perform abstractly the same task as simple sensors in the sense that they provide data upon an external request but consist in general of a number of different components". We argue that, if we consider EPMs to be such virtual sensors, they should also produce results comparable to physical sensors. Observations of physical sensors are made accessible using the OGC Sensor Observation Service (SOS) for the service interface and O&M for the encoding. Hence, we propose to wrap the output of EPMs using O&M. The O&M observation contains informative metadata and a reference to the resulting data. The latter, e.g. a coverage representing temperature variation, is delivered in one of the accepted spatial data formats such as GeoTIFF. Metadata about the EPM is included in the SensorML description referenced from the O&M document. Quality information can be directly nested in O&M using ISO 19115 or other more specific encodings such as the Uncertainty Markup Language (UncertML) (Williams et al, 2009).

Following this proposal, the SOS would be the most apparent choice for exposing EPMs. This could enable seamless access for generic SOS clients. However, this is problematic not only from a conceptual point of view. A SOS is meant to deliver observations and not to execute environmental models. In addition, models usually require specific input parameters which have to be provided by a human user (the unconfigured models discussed in the next section). The SOS does not support model-specific input and thus cannot be used here. Our architecture presents a solution which manages to keep support for SOS clients and exposes the EPMs as OGC Web Processing Services.

3.3.   Pre-configured vs. Unconfigured Environmental Process Models

Making EPMs accessible to a greater public requires means to let anyone invoke the composition without the need to understand the underlying (often very complex) configuration. The EPMs deliver, as suggested before, O&M documents with a reference to a SDS serving the actual data. Hence, EPMs are on first sight best exposed as SOS to enable the integration of generic SOS clients.

Pre-configured EPMs do not depend on any user interaction for their invocation. Input data is delivered by other Web services, whose location is preconfigured in the EPM. Calling this model will then fetch the data covering the requested time and area from the configured source. In addition, the standard SOS query parameters (e.g. to specify a bounding box for the desired coverage) are supported and bound to the internal variables of the EPM. Purpose and semantics of these parameters are specified in the SOS standard. Only in this case, service requests can be automatically generated by the SOS client software. A pre-configured EPM can be simply invoked by existing SOS-compliant clients if they rely only on standard SOS query parameters. Clients requesting this EPM don't have to be manually adapted.

Many environmental models, however, depend on input parameters. These parameters have to be provided by human users using specially adapted client software. In this case, we refer to the environmental models as being *unconfigured.* Some of the input parameters cannot be defined internally in the EPM; they have to be manually specified by the end user. Invoking the model without giving this additional information results in an error. A SOS, however, only accepts a limited set of parameters (i.e. spatial and temporal coverage or a sensor id). The WPS provides a more generic solution for migrating models into the Web. This flexibility comes with a serious limitation: whereas the SOS can be simply used in existing applications, the WPS almost always requires specially adapted clients due to its flexibility regarding the input parameters. Still, it is the only option for transforming unconfigured EPMs into OGC Web services. Similar to the proposal for pre-configured EPMs, the execution via a WPS interface results in an O&M document containing the modeled observations or a reference to a WFS/WCS serving large data sets. As EPMs are often quite complex and may require some time to be executed, manual invocation requires asynchronous communication for the model execution. The WPS specification includes extensions for the asynchronous communication to request the workflow execution state. The SOS is not meant to execute any processes; asynchronous data exchange is not specified for this service interface.

As conclusion we argue that pre-configured EPMs should be *accessible* through an OGC SOS interface to enable support by generic SOS clients. However, all

EPMs, regardless of their need for manually specified input parameters, should be *implemented* as WPS to support, for example, model-specific input parameters and asynchronous communication. Access via the SOS interface is managed by a SOS proxy which mediates client requests for pre-configured models from generic clients to the WPS implementation (see Section 3.6).

3.4. Environmental Models as Workflows

We presented the approach for migrating environmental models to the Web in a workshop with modeling experts (Urvois and Berre, 2009). The domain experts were mineral resource specialists from the French Geological Survey BRGM, who are responsible for the acquisition, compilation, and publication of geological data. GIS software is used for these tasks and the experts acquired in-depth knowledge about required data and processes. Moving their tasks into the Web has been recognized as useful at the workshop. But the idea of moving the whole workflow (acquisition, compilation, publication) was considered to be problematic. Without knowing what is going on inside the workflows, the users felt being "remotely controlled". They considered models as "Web-based black box", which made it difficult to be confident about the EPM results (Urvois and Berre, 2009). Making information about the EPM execution available is crucial to build up confidence.

Inspecting the model's behavior with respect to changes of the input data is therefore one key requirement for modeling experts. We distinguish between two options to publish computer models as services. Existing computer models can either be simply migrated as a whole into the Web. The configuration is then hard-coded into the implementation. Alternatively, they are partitioned into individual components (representing either data sources or model processes), with each module being exposed as Web service separately. EPMs represent the original model workflow and are themselves accessible as Web service. The first approach contradicts with the modularity and flexibility recommended for the GEOSS and INSPIRE architectures. The environmental model is bound to one specific scenario with a predefined spatial and temporal coverage. Being able to adapt environmental models requires loosely-coupled workflows. EPMs are configured under certain assumptions derived from the model builder's context.

Loosely-coupled BPEL workflows have the possibility of being translucent. ISO 19119 includes a distinction between transparent, translucent, and opaque model workflows (Percivall, 2002). Transparent workflows are configured and executed by humans. Translucent (or white-box) compositions are managed by workflow execution environments, but the users are still aware of the individual services. In the case of opaque (or black-box) models, the workflow's inner structure is hidden from the user (Einspanier et al, 2003). Translucent models can be easily communicated to the end user. They are able to reconstruct the model execution

by inspecting the performed steps and derive information about the provenance. The use of standardized workflow languages like BPEL helps to communicate the model structure and configuration. However, it does not answer which concrete parameter values were used. Information about the model execution is required here. This could be, for example, as simple as providing log files with the result. Letting users retrace the decisions taken during execution and inspect the content of flowing data requires a more sophisticated approach. The Open Provenance Model (OPM) is a mature standard for encoding and communicating lineage of computer models (Moreau et al, 2008). Hiding the model's structure and execution details from the user can sometimes be appropriate, too. For example, security constrains may force data providers to limit access to the workflows.

3.5. Architecture

In the preceding sections, we highlighted requirements for accessing and implementing EPMs on the Web. In the following, we specify the architecture which supports seamless integration with generic SOS clients, while using the WPS for the actual execution of the environmental model. Figure 1 summarizes the different points raised in this section. The *OGC Frontend to the Workflow Engine* is a service which exposes one or more EPMs as Web services. The models are implemented using the BPEL language and are deployed to and managed by the *Workflow Engine*.

Figure 1: OGC-compliant Frontend to the Workflow Engine
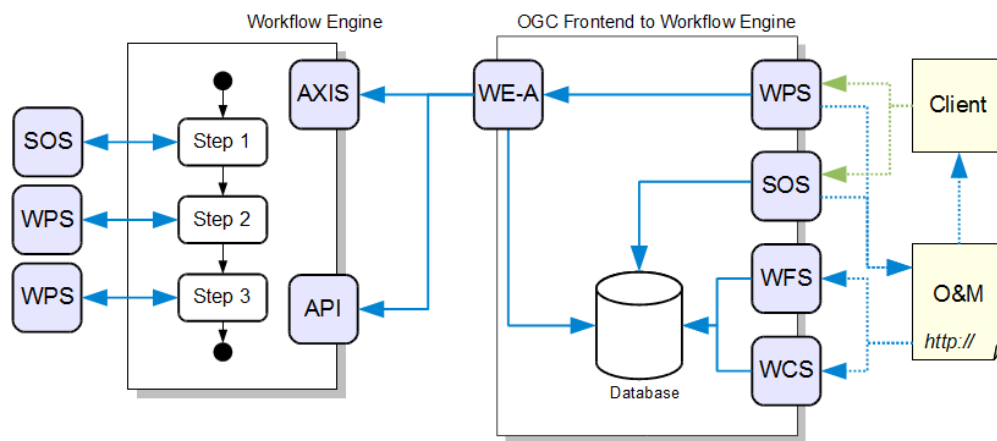
A generic client, for example a web-based map viewer, invokes the EPM by requesting environmental observations for a certain region and time using the SOS interface. If the observations are present in the database for the requested time and area, an O&M document with model metadata and the reference to a

SDS (WFS or WCS) is returned. The O&M-compliant client then simply fetches the data from the SDS. If the observations are not present, a simple feature of the underlying HTTP protocol is used. The SOS responds with a HTTP code 303 ("See other") with a pointer to the WPS responsible for executing the workflow producing the required spatial data. All input parameters from the original SOS request are forwarded to the WPS. In the case of a pre-configured model, the workflow expects only the usual SOS parameters as input. It is automatically executed; the response is either a HTTP code 202 ("Accepted, but not completed") if the output is not ready yet, or the O&M document with the requested information. For unconfigured models, further input parameters are required. Hence, the request with SOS parameters only results in an exception listing the missing parameters. It is then in the responsibility of the client to let the user specify these parameters and re-execute the WPS with all required inputs.

4. SCENARIOS

In the following section, we are going to present computer models for two real world scenarios. The first illustrates how preconfigured models for air pollution benefit from Web services. The second introduces a computer model for oil spills (and response actions) as example for unconfigured models.

4.1. Modeling Exposure to Air Pollutants

The raised awareness of the effects of air pollutants on human health lead to policies like the European directive on ambient air quality and cleaner air for Europe (DIRECTIVE 2008/50/EC). Determining air pollutant exposure to citizens is a challenging and unsolved task. The scenario described in this section introduces an environmental model addressing this issue. Parts of this scenario are also included in the Air Quality Health scenarios of the third GEOSS Architecture Implementation Pilot Phase 3 (AIP-3).

A non-ICT expert wants to explore the exposure of his children to air pollutants on the way to school and back. He decided to equip his children with GPS receivers. The resulting tracks are used to estimate their exposure to air pollutants. Figure 2 shows the workflow of this scenario. GPS data containing the tracks of the children to school and back home is first requested from a SOS. The availability of air quality information for the observed locations is checked next. A SOS providing air quality measurements extracted from the AirBase database of the European Environmental Agency (EEA) is used in this step. If concentration measurements are available for the requested positions, the workflow is finished and the concentrations are returned. If not, the workflow is split up into two sub-workflows: (i) including interpolation of the EEA rural background air pollutant measurements and (ii) estimating additional urban concentrations from local emission sources through an air pollutant dispersion model service. By adding

the two service responses, total air pollutant concentration estimates can be calculated for the requested GPS positions.

Figure 2: Workflow of the Air Quality Scenario
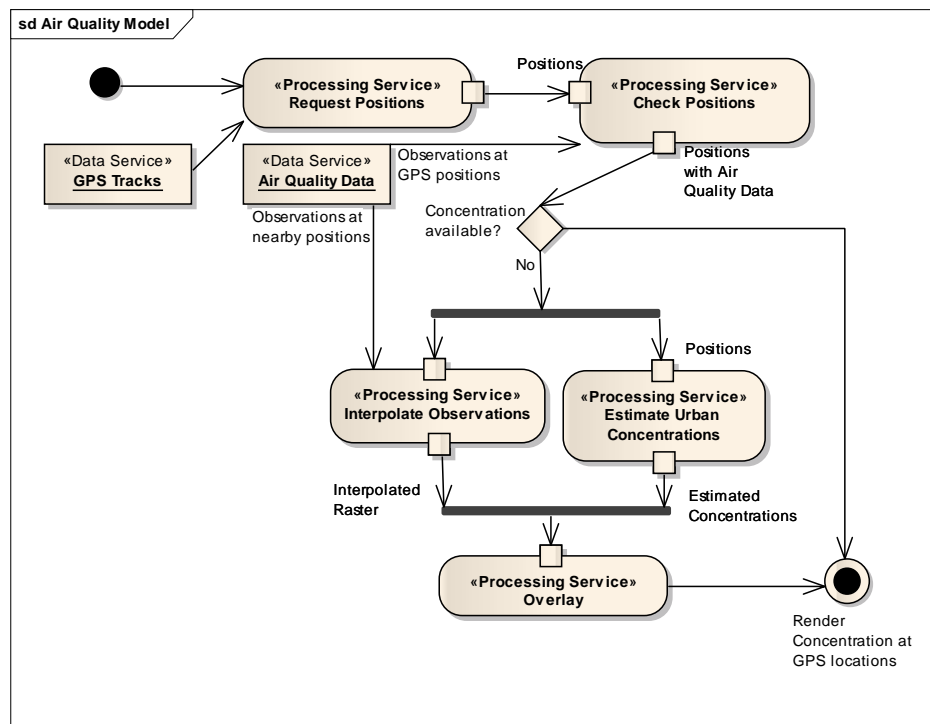
The complete workflow is encapsulated by a WPS. One of its internal Web services executes the AUSTAL2000 model, a local air pollution dispersion model developed for Germany. A second WPS offers interpolation functionality for the background concentration of air pollution. Its implementation, the INTAMAP interpolation service (Pebesma et al, 2009), relies on air quality observations for the respective time period. All sources for the input data, including the SOS reference to the tracks of the children, are predefined in this EPM.

4.2. Modeling the Impact of Oil Spills on Cod Populations

An early response is crucial to minimize the impact of oil spills on the ecosystem. Knowing how to fight the oil relies on computer models predicting the oil drift (and when it will reach the coastline). Accurate models of oil spill effects on the environment rely on additional information about the spill. This includes, amongst others, data about the nature of the oil spill (for example an oil well blow-out or a damaged ship loosing fuel), the spill location, the type of the spilled oil, the

water's temperature, salinity, and depth of the spill, direction and strength of wind and water currents, and others. A computer model already configured for one oil spill cannot be applied to another spill. It has to be reconfigured to accommodate the different conditions of the new event. The OSCAR (Oil Spill Contingency and Response) model supports the objective assessment of oil spill response strategies (Reed, 1995). In the ENVISION project we are searching for means to move parts of OSCAR into the Web. The scenario is concerned about the impact of oil spills on cod populations. It consists of two models: *Predict Oil Drift* models the oil distribution in a three-dimensional water body. An oil weathering model is then included to assess the impact of current weather and sea conditions on the oil's chemical composition and accordingly its behavior in seawater. Its output is required to predict the trajectory of the oil cover to assess when the oil cover reaches the coastline and eventually protected areas. The second model *Predict Cod Effects* estimates the toxicity (and subsequent lethality) for cod populations. The results of the oil drift prediction serve as input for modeling the uptake of oil components by cod eggs and larvae. Forecast data about sea and weather conditions is served by OGC-compliant Web services. The oil drift prediction model requires additional input parameters defining the chemical composition of the oil (crude oil, for example, behaves different in sea water then refined oil), the location of the incident, and the amount of oil discharge. This information cannot be sensed automatically, but has to be manually defined before executing the model. Hence, this scenario serves as an example of an unconfigured model that has to be adapted before being used in the decision-making process. Users have to manually specify certain input parameters in the client software before running the model (e.g. using a portal-based solution with map viewer as it is developed in ENVISION).

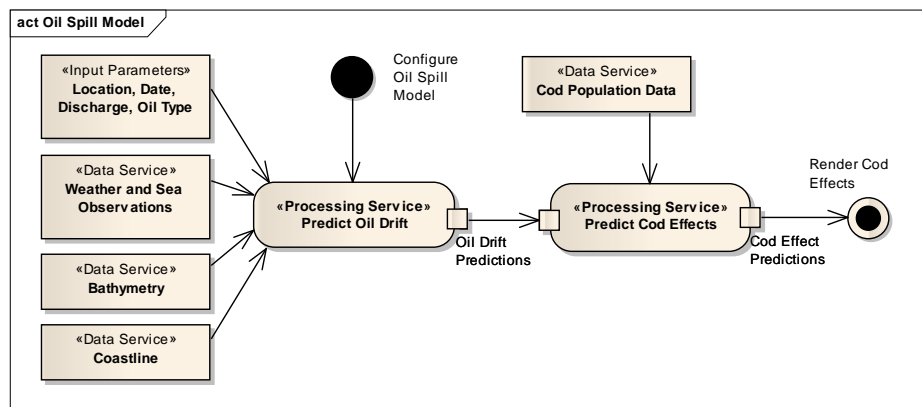**Figure 3: Workflow of the Oil Spill Model**

Figure 3 depicts the workflow of the introduced oil spill scenario. We can distinguish between three types of Web services delivering either near real-time data streams of (modeled) sensor observations (SOS), traditional spatial data sets (WFS, WCS), and processing capabilities (WPS). Additional parameters such as oil spill rate or oil type are provided by the application invoking the workflow, e.g. a web-mapping client.

5. DEPLOYING MODELS IMPLEMENTED AS WORKFLOWS

A BPEL workflow is a composition of activities, and/or other processes, with groundings for each activity through partner links. These links point to external services performing the activity tasks. The BPEL workflow itself is exposed as Web service again. Despite its widespread use, BPEL received substantial criticism mainly due to its strong link to WSDL (Christensen et al, 2001). Partner services are required to have a WSDL- based description according to the specification. However, OGC Web services (and also, for example, RESTful services) usually do not provide WSDL descriptions of their functionality. A workflow engine (as depicted in the architecture in Figure 1) is required for BPEL execution. A plethora of engines, either commercial (e.g. Oracle BPEL Process Manager[7]) or open source (e.g. Apache ODE[8]), are available. BPEL Workflow Engines were designed for executing business process models. Their support for other process model types, e.g. scientific process models, lags behind. In the following section, we list some of the requirements of environmental models which contemporary workflow engines still lack support of.

5.1. Environmental Process Requirements

Environmental process models pose significant requirements to the workflow engine, which are listed below:

- *Compliance to standards:* BPEL requires a definition of a WSDL Web service interface that is used to invoke the workflow. Integrating the workflows into existing spatial applications based on OGC standards requires an OGC WPS interface to the Workflow Engine (see Figure 1). The architecture presented acts as bridge between the WPS and the WSDL interface exposed by the workflow engine.
- *Loosely-coupled processes:* Environmental process models comprise spatial data and processes offered by distinct providers. The interactions between the process model and the data or processing services have to be performed in a peer-to-peer manner supporting loose coupling of the individual activities. Both, the process model and the providers, remain autonomous during the performed interactions.

---

[7]   Oracle BPEL Process Manager: http://www.oracle.com/appserver/bpel_home.html
[8]   Apache ODE: http://ode.apache.org/

- *Scalability in terms of supported process instances and data volume:* Deploying environmental process models increases significantly the number of process instances. Process models work with information ranging from simple text to raster images and extensive feature collections. Existing spatial data services already struggle to deal with voluminous data, processing the data across many running instances poses a significant scalability concern for the deployment infrastructure.
- *Transfer of spatial data:* Due to bandwidth and/or network throughput limitations, the exchange of voluminous data delays process model execution. Filtering may enable to limit the exchanged data and therefore network load to a certain extent.
- *Environmental model performance:* Clients loading environmental model results into a map view expect fast visualization of the results. The complexity of the spatial operations performed by the partner services, as well as the exchange of voluminous data, sets significant performance requirements on the deployment infrastructure.
- *Use of multiple types of services:* Environmental process models comprise interactions with different processing and data providing partners. Those are accessible using distinct types of services. For example, processing functionality may be offered through either W3C- or OGC-compliant services.

Some of these issues are addressed neither by BPEL nor by the contemporary BPEL engines. Web-enablement and peer-to-peer interaction are inherently supported by BPEL engines. The interaction between engine and partner services is performed in a point-to-point manner. Scalability, transfer, performance, and the use of multiple service types remain challenging. BPEL has been primarily conceived as a mechanism facilitating the provision of business processes. Here, the exchange of voluminous data or varying service types was not considered relevant. Scalability, transfer, and performance are mainly related to the architecture, e.g. choosing a distributed or centralized system, and the deployment infrastructure, e.g. support for multithreaded and/or parallel execution. The other issues are related to the BPEL specification, in particular the support of different service types. This can be achieved either through extending BPEL as proposed by Karastoyanova et al (2008). Tsalgatidou et al (2008) suggest to use appropriate WSDL bindings for the additional services types. OGC services have to be supported for environmental models. This can be natively addressed by BPEL using the WSDL HTTP binding mechanism.

5.2.   Oil Spill BPEL Process Model

The oil spill model is an unconfigured environmental process model depending on additional parameters for its invocation. The location and time of the oil spill as well as the amount and type of the oil have to be specified by the modeling expert. The result of the workflow (as depicted in Figure 3) contains predictions of the oil spill's effect on cod populations. We mentioned that unconfigured models

have to be exposed as WPS processes. The following Listing 1 is a simplified WSDL example of the oil spill workflow (in this case we assume that one WSDL file is created for each process). The WSDL example does not specify concrete bindings to a service instance. Details such as namespaces, mandatory attributes, or operations like the getCapabilities-method have been left out as well[9].

Listing 1: Excerpt of the WSDL Interface for the Predict Oil Drift-Model

```
1  <definition name="ExposedWPS">
2
3   <!-- TYPES -->
4   <types>
5    <xsd:element name="Execute_Request" type="tns:Execute_RequestType"/>
6    <xsd:complexType name="Execute_RequestType">
7     <xsd:sequence>
8      <xsd:element name="SpillLocation" type="gmls:CartesianCSType"/>
9      <xsd:element name="SpillDate" type="xsd:date"/>
10     <xsd:element name="OilType" type="xsd:string"/>
11     <xsd:element name="OilDischarge" type="xsd:float"/>
12    </xsd:sequence>
13   </xsd:complexType>
14   <xsd:element name="Execute_Response" type="tns:Execute_ResponseType"/>
15   <xsd:complexType name="Execute_ResponseType">
16     <xsd:sequence>
17       <xsd:element name="Execute_Result" type="tns:FeatureCollectionType"/>
18     </xsd:sequence>
19   </xsd:complexType>
20  </types>
21
22  <!-- MESSAGES -->
23  <message name="Execute_RequestMessage">
24   <part name="parameters" element="tns:Execute_Request" />
25  </message>
26  <message name="Execute_ResponseMessage">
27   <part name="body" element="tns:Execute_Response" />
28  </message>
29
30  <!-- PORTS -->
31  <portType name="WPS_PortType">
32   <operation name="Execute">
33    <input name="Execute_input" message="tns:Execute_RequestMessage"/>
34    <output name="Execute_output" message="tns:Execute_ResponseMessage"/>
35   </operation>
36  </portType>
37
38 </definition>
```

At line 32, the operation execute() is implemented according to the WPS specification. The input of this operation (l.33) has one part (l.24) which is further defined in the section of WSDL Types (l.5-12). Here, the mandatory input parameters are specified. This example illustrates how unconfigured models are mapped to WSDL interfaces. The mandatory information for executing

---

[9]   The source code of this example is part of the ENVISION project, and can be found in its source code repository at http://kenai.com/projects/envision.

unconfigured models are linked to WSDL input parameters, whilst the model result is mapped to the output type of the operation. In this case, the result would be a Feature Collection. Similarly, pre-configured models can be mapped to operations without input parameters. The WSDL file presented here is the service description exposed by the workflow engine. The OGC Frontend offers access to the traditional OGC-compliant service capabilities.

Listing 2: Simplified Example of the BPEL
Process for the Predict Oil Drift-Model

```
1  <process name="OilSpillModel">
2   <sequence>
3    <!-- Get request with parameters, assign to variable -->
4    <receive operation="ModelRequest" variable="Model_Input"
5            partnerLink="ExposedWPS"/>/>
6
7    <!-- copy input parameter to WPS input variable -->
8    <assign>
9     <copy>
10     <from>$Model_Input.parameters/SpillLocation</from>
11     <to>$WPSRequest.parameters/Location</to>
12    </copy>
13   </assign>
14
15   <!-- Invoke remote WFS, assign response to variable-->
16   <invoke operation="getFeature" partnerLink="WFS"
17           outputVariable="RemoteResponse"/>
18
19   <!-- copy response to input of next service -->
20   <assign> .. </assign>
21
22   <!-- repeat for other input services  -->
23
24   <!-- Invoke WPS for predicting oil drift-->
25   <invoke operation="execute" partnerLink="WPS"
26           inputVariable="WPSRequest"
27           outputVariable="WPSResponse"/>
28
29   <!-- assign output to model response variable -->
30
31   <!-- return computed result to model client-->
32   <reply operation="execute" variable="ModelOutput"
33           partnerLink="ExposedWPS"/>
34
35  </sequence>
36
37 </process>
```

OGC services rely on HTTP-based (e.g. HTTP GET or HTTP POST) interaction patterns. Using the appropriate WSDL binding type is crucial for their integration in BPEL processes. BPEL editors support the transformation of graphical models as in Figure 3 into executable BPEL process specifications such as the one presented in Listing 2. In this simplified sequence, a remote WFS (representing

an input data source) and a WPS (representing the model process) are invoked. The incoming request (l.4) is using the execute operation defined in the WSDL file (the PartnerLink *ExposedWPS* maps to the WSDL file in Listing 1). The input variable *Model_Input* includes the mandatory parameters which are copied into the request for the remote WPS (l.7-13). The same happens for the WFS whose response for the getFeature operation is copied into the variable RemoteResponse (l.15-20). The WPS providing the model process is invoked with all input data (l.24-27) and its result is returned as result of the whole workflow (l.31-33).

6. EVALUATION

Various related work addresses the question how to integrate and task models. Klopfer and Simonis (2009) implemented, for example, an interpolation Web service. Similar to the approach presented, they understand the model as a virtual sensor and publish it as Sensor Planning Service (SPS) (Simonis, 2007). However, in our experience, the WPS offers all functionality needed to execute environmental models and the WPS is already established for providing processing functionality in SDIs. A service-based application for alpine run-off models has been introduced by Granell et al (2010). It consists of a portal application, a service layer, and a data layer. The portal application allows for the discovery of available services, the visualization of its data, and the orchestration of services within environmental models. The models themselves are executed by the portal application. The service layer provides means to integrate external SDS. This enables access to several data sources (in the data layer) like maps, coverages or feature-based spatial data. The presented approach goes beyond the related research by combining standards for the Sensor Web with already well-established geospatial services like the WPS, WCS, or WFS.

We applied a WPS-based approach for publishing environmental models in the projects GDI-Grid, SWING, and INTAMAP. User requirements are crucial to understand the choice for SOS/WPS coupled with a WFS or WCS. Even though technically simple to implement and adapt, the WPS standard is rather vague regarding its output and input. Implementations are prone to have conflicting interfaces. Seamless integration into scientific workflows by coupling and comparing the models is nearly impossible to achieve if the WPS interface is used without further constraints (for example through profiling). Since models are supposed to approximate observations of a (usually future) reality, we argued that clients best deal with environmental models in the same way as (virtual) sensors. In some cases, though, environmental process models rely on input

parameter which cannot be retrieved automatically. Here, the WPS is the most appropriate choice[10] .

We have also evaluated different approaches for executing environmental process models: In SWING, we have taken an approach based on abstract state machines. Its performance did unfortunately not match the user requirements; one execution took around four minutes. In addition, this approach was too complex and required specific knowledge about the chosen composition language to adapt the workflows for different scenarios. In the GDI-Grid project, we investigated the use of parallel processing using grid technology to enhance performance of the workflow execution. Kurzbach et al (2009) deployed numeric flood models and Lanig et al (2008) investigated the use of the WPS for processing digital terrain models. In the GDI-Grid and the SODIUM project, BPEL and BPEL-like languages respectively have been successfully tested. Tsalgatidou et al (2006) suggest that BPEL-like declarative languages can easily support the provision of scientific processes comprising heterogeneous types of services. Moreover, BPEL and BPEL-like languages provide the necessary level of abstraction which supports the deployment of scientific processes over advanced platforms. In the ENVISION project we combine these results to enable efficient execution of service compositions. The issues discussed about semantics arose (and have partly been dealt with) in GDI-Grid and SWING. The approach based on semantic annotations has been successfully tested for the discovery of Web services and the validation of service compositions. Within the INTAMAP project, interpolation algorithms were served through a WPS interface (Pebesma et al, 2009). The XML language developed for encoding uncertainty, UncertML, has been proven to be appropriate to encode data quality aspect within the metadata of spatial data. Further integration of uncertainty into workflows in the Model Web is the subject of the UncertWeb project. The concerns of end users about the lack of confidence in results of opaque models have been recognized and are also addressed in ENVISION and UncertWeb.

7. CONCLUSION

Standards for the Web-enablement of environmental models were discussed. Some of these models can be considered to be virtual sensors offering estimated measurements for physical phenomena in the real world. The OGC standards for encoding sensor observations enable the integration of environmental process models into existing SDIs. Workflow examples for two scenarios from the domain of air quality and oil spills were presented. The former serves as example for a non-interactive and pre-configured model. The latter is an unconfigured model due to the dependency on manually defined input parameters. The sub-

---

[10]   Initially the SPS was considered here, but was later discarded. The WPS largely overlaps with the SPS and is more established in the geospatial community.

processes of the model workflows, e.g. interpolation, numeric prediction, or topological operations, are each encapsulated as OGC WPS. The complete workflows are published as services again to ensure reusability. Standardized workflow languages support translucent models to let users study the model's inner workings.

Since environmental models are considered to be virtual sensors, their results are encoded using the O&M format. This has the following advantages: (i) O&M provides a common element to describe the procedure that has produced the observation's (or in our case model's) result. Hence, model metadata can be referenced from the model output in a common way. (ii) O&M provides common elements to describe the quality of results. The INTAMAP project already successfully explored how to integrate uncertainty information into this model. (iii) By explicitly distinguishing observation metadata and observation result, it is possible to provide the model results through the best-suited formats and service interfaces (e.g. NetCDF and WCS for model results which are coverages) together with a common way to describe the metadata (see (i) and (ii)). This can be achieved by returning the model outputs as observations and referencing the data services like WCS and WFS from these observations. However, the general O&M model is quite generic and defines a lot of optional elements or elements without type restrictions. Thus, the definition of domain specific profiles like hydrology or meteorology might further increase interoperability.

The WPS is the best choice for the execution of environmental models. It offers common ways to define parameters needed for the execution and a common way for asynchronous communication. Integrating a generic WPS into existing applications is difficult. Calling a WPS almost always requires some sort of client-side implementation which contradicts the presented vision of seamless integration of environmental models. This is supported by the SOS, which should be selected for accessing pre-configured environmental models. It helps to avoid the complexity of invoking environmental models, but still gives the user the opportunity to inspect the model structure. The workflows are encoded using the established BPEL standard and can be executed using generic BPEL execution engines. The approach presented should be considered as a compromise. The lack of an OGC service specification covering the needs of unconfigured models should be on the agenda of future OGC efforts. The present interoperability technology can only cover a small part of the existing desktop-based environmental models. Climate models, for example, are far too complex to be deployed in Web service architectures. The number and variability of the input parameters, their often unforeseeable impact on the model results, and the expertise required to understand and re-trace this impact requires highly interactive and specialized applications. Only certain steps of the models may be migrated to benefit from Web-enablement, e.g. to support faster execution through distributed computing.

The Model Web is still under development and can be brought forward only by deploying models on the Web. Several other open issues have to be further investigated to enable seamless migration of existing environmental models. One important fact is the communication of semantics and uncertainty to ensure interoperability and extensibility of environmental models. Ontologies that capture the dynamics of geographic space can establish a link between sensors observing real-world geoprocesses and environmental models simulating these sensors with the ultimate goal to seamlessly switch between both. The quality of the resulting information, in particular its uncertainty, has to be considered before integrating models in potentially critical applications. And in the end, the user has to be aware of the model concept, its uncertainty, and the underlying assumptions. A user should be able to evaluate the resulting data, including selected data sources and parameters, a description of the applied algorithm, and the quality of produced result. The proposed standards in this paper support the communication of such model aspects and are therefore the best choice to expose environmental models as Web service.

ACKNOWLEDGEMENTS

REFERENCES

Athanasopoulos, G., Fox, E., Ioannidis, Y., Kakaletris, G., Manola, N., Meghini, C., Rauber, A. and D. Soergel (2010). "A Functionality Perspective on Digital Library Interoperability", Proceedings of 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2010), Sept 6-8, Glasgow, UK.

Botts, M. and A. Robin (2007). OpenGIS Sensor Model Language, OGC 07-000, Open Geospatial Consortium, Inc.

Christensen, E., Curbera, F., Meredith, G. and S. Weerawarana (2001). Web Services Description Language (WSDL) 1.1, World Wide Web Consortium (W3C) Note, 15 Mar. 2001, at: http://www.w3.org/TR/wsdl [accessed 15 February 2011]

Christian, E. (2005). Planning for the Global Earth Observation System of Systems (GEOSS), Space Policy 21 (2), pp 105–109.

Cox, S. (2007). Observations and Measurements Part 1 - Observation Schema, OGC 07-022r1, Open Geospatial Consortium, Inc.

Diane, J. and J. Evdemon (2007). Web Services Business Process Execution Language Version 2.0, OASIS Standard.

Einspanier, U., Lutz, M., Senkler, K., Simonis, I. and A. Sliwinski (2003). "Toward a Process model for GI Service Composition", Proceedings of GI-Days 2003, June 26-27 2003, IFGI, Münster, Germany.

Frigg, R. and S. Harmann (2006). Models in Science, at http://plato.stanford.edu/entries/models-science/, [accessed 14 June 2010].

Geller, G. N. and F. Melton (2008). Looking Forward: Applying an Ecological Model Web to assess impacts of climate change, Biodiversity 9 (3&4).

Granell, C., L. Díaz and M. Gould (2010), Service-oriented Applications for Environmental Models: Reusable Geospatial Services, Environmental Modelling & Software 25 (2): pp. 182–198.

Highland, H. J. (1973). A taxonomy of models, ACM SIGSIM Simulation Digest 4 (2) pp. 10–17.

Karastoyanova, D., Van Lessen, T., Leymann, F., Nitzsche, J. and D. Wutk(2008). WS-BPEL Extension for Semantic Web Services (BPEL4SWS) Version 1.0.

Klopfer, M. and I. Simonis (2009). SANY - An Open Service Architecture for Sensor Networks, SANY Consortium.

Kurzbach, S., Pasche, E., Lanig, S. and A. Zipf (2009). Benefits of Grid Computing for Flood Modeling in Service-oriented Spatial Data Infrastructures, GIS.Science, 3/2009: pp. 89-97.

Lanig, S., Schilling, A., Stollberg, B. and A. Zipf (2008). "Towards Standards-Based Processing of Digital Elevation Models for Grid Comput- ing through Web Processing Service (WPS)", ICCSA '08: Proceedings of the International Conference on Computational Science and Its Applications, Part II, Springer, Berlin, Heidelberg, pp. 191–203.

Maué, P. and S. Schade (2009). Data Integration in the Geospatial Semantic Web, Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications, 11 (4): pp. 100–122.

Mineter, M., Jarvis, C. and S. Dowers (2003). From Stand-alone Programs Towards Grid-aware Services and Components: A Case Study in Agricultural Modelling with Interpolated Climate Data, Environmental Modelling & Software, 18 (4): pp. 379–391.

Na, A. and M. Priest (2007). OGC Sensor Observation Service Implementation Specification, OGC 06-009r6, Open Geospatial Consortium, Inc.

Pebesma, E., Cornford, D., Dubois, G., Heuvelink, G., Hristopoulos, D., Pilz, J., Stoehlker, U. and J. Skoien (2009). "INTAMAP: An Interoperable Automated Interpolation Web Service", Proceedings of StatGIS 2009, June 16-18 2009, Milos, Greece.

Percivall, G. (2002). "ISO 19119 and OGC Service Architecture", Proceedings of FIG XXII International Congress, Washington D.C., USA.

Reed, M. (1995). Quantitative Analysis of Alternate Oil Spill Response Strategies using OSCAR, Spill Science & Technology Bulletin 2 (1): pp. 67–74.

Schut, P. (2007). OGC Web Processing Service Implementation Specification, OGC 05-007r7.

Sgroi, M., Wolisz, A., Sangiovanni-Vincentelli, A. and J. Rabaey (2005). "A Service-Based Universal Application Interface for Ad Hoc Wireless Sensor and Actuator Networks", in: W. Weber, J. Rabaey, E. Aarts (Eds.), Ambient Intelligence, Springer.

Simonis, I. (2007). OpenGIS Sensor Planning Service Implementation Specification, OGC 07-014r3, Open Geospatial Consortium, Inc.

Tsalgatidou, A., Athanasopoulos, G., Pantazoglou, M., Pautasso, C., Heinis, T., Grønmo, R. , Hoff, H., Berre, A., Glittum, M. and S. Topouzidou (2006). Developing Scientific Workflows from Heterogeneous Services, SIGMOD Rec. 35 (2): pp. 22–28.

Tsalgatidou, A., Athanasopoulos, G. and M. Pantazoglou (2008). Interoperability Among Heterogeneous Services: The Case of Integration of P2P Services with Web Services., International Journal of Web Service Research 5 (4): pp. 79–110.

Urvois, M. and A. J. Berre (2009). D1.3 - Experience Report, SWING Project Deliverable.

Vretanos, P. (2005). Web Feature Service Implementation Specification, OGC Implementation Specification, OGC 04-094Open Geospatial Consortium, Inc.

Whiteside, A. and Evans, J. (2007). Web Coverage Service (WCS) Implementation Standard. OGC 07-067r5, Open Geospatial Consortium, Inc.

Williams, M., Conford, D., Bastin, L. and E. Pebesma (2009). Uncertainty Markup Language (UncertML), OGC 08-122r2, Open Geospatial Consortium, Inc.