The ENVISION Environmental Portal and Services Infrastructure

Patrick Maué[1] and Dumitru Roman[2]

[1] Institute for Geoinformatics (ifgi), University of Münster, Germany
patrick.maue@uni-muenster.de
[2] SINTEF, Oslo, Norway
dumitru.roman@sintef.no

Abstract. The ENVISION Portal is a Web-enabled infrastructure for the discovery, annotation, and composition of environmental services. It is a tool to create Web sites dedicated to particular domain-specific scenarios such as oil spill drift modeling or landslide risk assessment. The underlying architecture based on pluggable user interface components is briefly discussed, followed by a presentation of the components resulting from the first iteration of the implementation. A walkthrough explains how to create a scenario website and populate it with the user interface components required for one specific scenario. The paper concludes with a discussion of open challenges identified during the implementation.

Keywords: Environmental Services, Environmental Models, Portals, Ontologies

1 Introduction

Geospatial workflows such as the processing and analysis of environmental data to support decision making are traditionally realized with Geographic Information Systems (GIS). The benefits of distributing tasks into the Web have been acknowledged by the geospatial community. Since the INSPIRE directive entered into force, the number of geospatial Web services embedded in Spatial Data Infrastructures (SDI) has grown considerably. Standards by the Open Geospatial Consortium[3] (OGC) have been specified to enable seamless integration of these Web service into existing geospatial workflows. The standards specify Web service interfaces, data models, and encodings for feature- and raster-based data, or data coming from in-situ environmental sensors. Even though there has been considerable uptake of these Web services in the geographic information community, they are playing only a small role for environmental modeling infrastructures.

Environmental computer models, which, for example, predict future weather conditions or interpolate temperature observations, are still rarely seen on the Web. Migrating existing models into the Web, and the development of Web-based user interfaces for environmental models, is a challenging task. But the

[3] Website of the OGC http://www.opengeospatial.org/

outcome of this process is of large interest not only for the environmental modeling community [1]. Initiatives such as SEIS ("Shared Environmental Information System") and SISE ("Shared Information Space for Europe") call for a better availability of environmental information for the end users, both public authorities and citizens. Making environmental computer models available on the Web is also the vision of the "Model Web" [2]. The Model Web has been defined within GEOSS ("Global Earth Observation System of Systems") [3] as the potential solution to enhance the interoperability of environmental computer models. It is a vision of a multidisciplinary network of models, data sources, processes, and sensors, which seamlessly interact through the well-defined Web service interfaces specified by the OGC. The interoperable computer models are supposed to be available to researchers, managers, policy makers, and the general public. Environmental service infrastructures are one first step towards the idea of havings Models as services in the Model Web [4].

Several open issues contribute to the lack of environmental services on the Web. The existing models are often complex monolithic systems implemented decades ago in programming languages incompatible with modern distributed architectures. The variability of the input parameters is challenging, and the computer model's graphical user interface required to specify these parameters cannot be simply migrated into the Web. The required data sources are often locked away due to licensing issues and the often proprietary formats impair their reusability for other applications. Another open issue is the modeling expert's fear of losing control over the model execution. In a workshop with geologists working with the modeling infrastructure developed in the research project SWING, the experts were in general convinced about the benefits of migrating existing models into the Web [5]. But they also mentioned the importance of communicating how the model has been executed, e.g. by not also only delivering the resulting maps but also the results of intermediate steps and the input parameter and input data.

The ENVISION ("Environmental Services Infrastructure with ontologies") project[4] started early 2010 as follow-up project to SWING [6]. The research aims to enable the migration of computer models into the Web through the conceptual specification and implementation of an environmental services infrastructure. The results include tools for a semantically enhanced and multilingual discovery of existing environmental services and the adaptive composition of environmental models as services. The Web-enabled creation and publication of scenario Web sites built on these models supports decision making in the environmental domain even for users with little technical skills. The scenario Web sites visualize the results of the environmental models using interactive map viewers, diagrams, and other scenario-dependent components. A pluggable user interface based on *Portal* frameworks ensures a high flexibility for the configuration of these Web sites. This paper introduces the underlying architecture of Portals and the ENVISION user interface components.

[4] See `http://www.envision-project.eu`

The presented research builds on two scenarios to validate the implemented infrastructure for environmental service. Risk assessment for landslide hazards is subject of the first scenario; the focus lies on the integration of real-time data coming from in-situ sensors deployed in the pilot area (the French department Guadeloupe). In the second pilot, the impact of oil spills in the Norwegian Sea on the development of the cod population will be investigated. For both scenarios a set of OGC-compliant Web services coming from various organizations are re-used.

In the ENVISION project, we are following the SEIS principles by enabling end-users to participate in environmental decision making processes through the planned environmental services infrastructure. The provided open source technologies will support the easy creation of new environmental models by coupling existing services to compositions, and to publish these models in domain-specific scenario Web sites which enable both, the general public and the environmental experts, to interact with the models and integrate them into decision-making processes.

This paper gives the reader a first impression about the ENVISION Portal. We start with a short overview of the general architecture, and then walk through the steps required to setup an environmental model and publish it on a scenario Web site. Before we give an outlook in the conclusion, we discuss open challenges which emerged during the first iteration in the development of the platform.

2 Architecture

The primary target of this research has been the specification and implementation of a Web-based portal which supports users in the discovery, annotation, and composition of new environmental models exposed as Web services. The portal enables the creation of domain-specific Web sites. These Web sites enable end users to interact with the models, preferably on maps augmented with interface components suitable for changing and calibrating certain aspects of the model. This approach requires a pluggable user interface, which makes it easy for the scenario Web site designers to select from a library of components useful for one particular computer model.

The user interface is based on Portlets based on the standard "JSR 286: Java Portlet Specification"[7]. The Portlets are the individual building blocks of a Portal. The latter is in its broadest sense an entry point to access certain domain-specific information. The purpose of portal software is similar to other Content Management Systems: it provides a framework for Web sites in need for personalization, authentication, and the management of the presentation layer of information systems [7]. The Portal is provided by the Portal Container, which manages the initialization and rendering of the portlets as well as Inter-Portlet Communication (IPC). Several open source implementations of Portal Containers compliant to JSR 286 exist, with Apache Jetspeed[5] and Liferay[6]

[5] See http://portals.apache.org/jetspeed-2

[6] See http://www.liferay.com

being probably the most popular choices. The Portlets are delivered through Portlet Repositories. The portal software linked to these repositories then manages the installation and updating of the deployed portlets. The portal consists of several portal pages, and each page may be composed of multiple Portlets. The Portlets itself communicate with back-end Web services, which perform most of the tasks required in environmental services infrastructures.

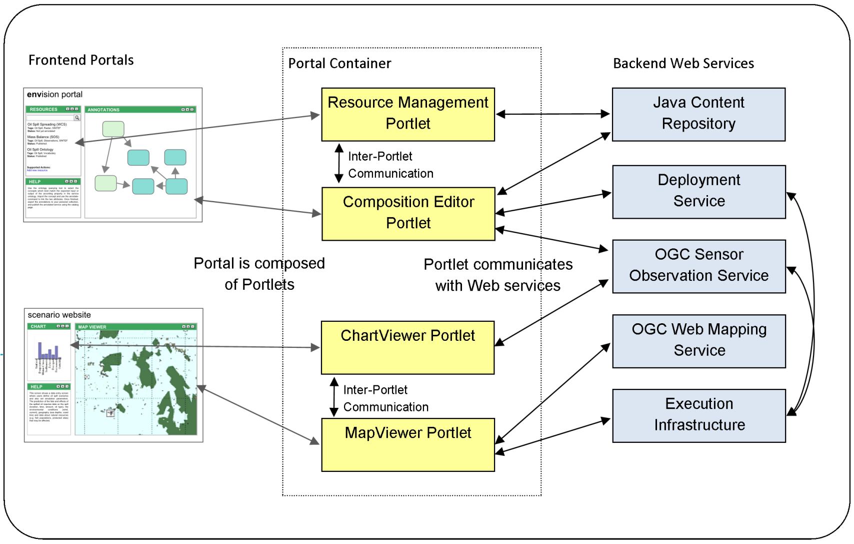


Fig. 1. High-level overview of architecture

Figure 1 illustrates the basic dependencies between the frontend Portals (the scenario Web sites and the ENVISION Portal), the Portlets, and the backend Web Services. The ENVISION Portal is an infrastructure to manage environmental Web services and to configure and publish scenario Web sites. The latter are domain-specific Portals, dedicated to one specific scenario such as showing landslide models for one particular region. The scenario Web sites are implemented as sub-portals, which are conceptually separated from the main Portal, but share the same platform. Creating a scenario Web sites involves the selection of layout, portlets, access rights, and more.

The Portals are composed of Portlets. The communication between portlets is realized through IPC events as defined in the JSR 286 standard. If, for example, a user selects a sensor in a portlet showing a map, another portlet on the same page automatically updates a displayed chart with the data of this specific sensor. A pluggable user interface relies on loosely coupled, but well defined interfaces. It should be easy to switch between different implementations for certain portlets. Loose coupling also decreases the risk of functional dependencies between the components. One portlet has to remain fully functional (but not necessarily useful) if other portlets are not present on the same page. The map's notification for the timeline to update itself therefore has to go through this interface. This technique can potentially work across pages and even different user sessions, making it for example possible to update maps on other users' browsers by the model designer if, for example, the model has been re-computed due to new input data.

2.1 Portlets for the Scenario Web sites

As illustrated in Figure 1, a Scenario Web site comprises Portlets used, for example, for interacting with a map (MapViewer Portlet) or a chart visualizing a time series of observation data (ChartViewer Portlet). Additional portlets, e.g. displaying static content explaining the purpose of the scenario, are shipped with the Portal Container software. This also includes Portlets for community building (e.g. Wikis, Forums, Chats), aggregators (display of RSS feeds, embedding of remote videos, Blogs), and many more.

The MapViewer Portlet (see Figure 4) as central entry point for the end users is built on OpenLayers[7], an open source javascript framework to visualize geospatial data on a base map. The library comes with built-in support for OGC Web Feature Services (WFS) and OGC Web Mapping Services (WMS, including WMS Time), and rudimentary functionality to interact with an OGC Sensor Observation Service (SOS). It lacks support for the Web Coverage Service (WCS), which means that coverages (in our case in the NetCDF format[8]) have to be channeled through an OGC WMS. The WMS transforms the coverage into a binary image which can be visualized by the client. In ENVISION, some of the services do not necessarily have an OGC-compliant interface. The composed workflows, for example, are at the moment deployed on standard workflow engines which only expose a Web service interface based on the W3C SOAP and WSDL standards. Hence, the Open Layers client had to be extended to also support these service interfaces.

Other crucial portlets for the scenario Web sites include the uncertainty-enabled ChartViewer Portlet (see also Figure 4) and the Notification Portlet. The former expects the sensor data encoded in the OGC Observation&Measurement [8] format to be augmented with uncertainty information. In this case, we rely on the UncertML standard [9]. In-situ sensors produce measurements within a certain confidence interval. If a gaussian distribution for the measurements can be assumed, the observation would include the mean as well as the upper and lower confidence intervals. The chartviewer renders the mean as expected observation value, and the confidence intervals to illustrate the uncertainty of this value. The notification portlet enables users to subscribe to certain events which

[7] See `http://openlayers.org/`

[8] See `http://www.unidata.ucar.edu/software/netcdf/`

can be associated to the models and sensor services visualized in the MapViewer Portlet. The nature of the event is modelled in domain ontologies. Complex Event Processing [10] is applied to extract the events from the data streams and populate the ontologies with event instances. The underlying architecture is quite complex, and explained in more detail in [11].

2.2 Portlets for the ENVISION Portal

The ENVISION Portal is separated from the scenario Web sites and comes with its own set of portlets required to manage and publish environmental models. The Portal is (for now) composed of three pages for the discovery, annotation, and composition of environmental models. The Resource Management Portlet acts as a bridge between the different pages and manages the user collection of Web services, Ontologies, and Composition Drafts. The user collection is shared across different Portals, the user's email address acts as unique identifier of the collection. The resources are stored in a repository compliant to the Content Repository API for Java [12], which supports the storage and retrieval of arbitrary documents in a tree-like structure.

The annotation consists of a set of portlets enabling the user to link existing Web service with the ontologies. The annotation procedure is based on the methodology developed for the SWING project. We distinguish between the ontology querying tool (using the *Ontology Query Portlet*) and the actual semantic annotation with the visualized ontology. Querying supports natural language queries (expecting the ontologies to be grounded with a rich set of documents) which enables users unaware of the specifics of the ontology to identify the correct concepts. The annotation screen then enables the user to select the entities in the data model (e.g. an attribute of an feature type coming from an OGC WFS) and link them to the discovered ontology concept.

The *Discovery Portlet* is backed by a standard OGC catalogue (in this case we are using the Geonetworks[9] implementation). The Discovery Portlet is a standard catalogue client with support for spatial and keyword-based queries. In the next step, it will be enhanced with semantic query capabilities. The user will be able to browse the ontologies and select the concepts and relations which best reflect the required data. The semantic query is encoded within a standard OGC CSW query. A semantic catalogue adapter is then responsible to forward the standard OGC query constraints to the existing catalogue, and the semantic part to a reasoning engine.

The *Composition Editor Portlet* allows user for coupling OGC Web services via workflows. The core processing of an environmental model (e.g. the interpolation method) are separated from the Web services delivering the input data (e.g. sensor data streams, feature collections, or coverages). With the composition the input data services are linked to the processing services, the result is a workflow document encoded in the Business Process Modelling Execution Language (BPEL) [13] which can be deployed on standard workflow engines such as

[9] See `http://geonetwork-opensource.org/`

Apache ODE [10]. The workflows are specified using the Busness Process Modelling Notation (BPMN) [14], and translated to executable BPEL specifications. The BPMN editor is based on the Oryx Editor[11], a popular Web-based open source BPM diagram editor for the Web [15].

The portlets are all backed up by Web services implementing the processing, execution, and resource management. In particular, the execution of the geospatial workflows can be challenging. The amount of the transported data can be massive, and the processing algorithms can be resource-intensive. A distributed execution infrastructure based on Peer-to-Peer technology is currently implemented in ENVISION to ensure efficient and semantically-aware execution of the workflows [16].

3 Setting up a Scenario Web site

The following walkthrough explains the steps required to compose a new environmental model from existing Web services and to publish it with a scenario Web site. The scenario is based on a simple use case developed for the ENVISION project. A WPS has been implemented to model the risk of drought in urbanized areas in France. A range of sensors measuring the current groundwater levels serve as input (via an SOS interface) for the classification. The features representing the urbanized areas come from a WFS loaded with the Corine landcover dataset. The activities described here also illustrated in a video[12].

The targeted user of the ENVISION Portal is the domain expert with knowledge about setting up an environmental model. This does not necessarily include any ICT skills. The Portal is supposed to enable experts with particular expertise in one particular environmental domain (e.g. oil spills) to create a Web site. He might know how to program a model, and has in depth knowledge about the expected outcomes of models and the impact of the input parameter. He most probably has no experience in Web services, OGC standards, data formats, or Ontologies. The end users, which are the target audience of the Scenario Web sites, might have neither experience in environmental models nor ICT. Here, the focus lies on a simple user experience, which enables any user to interact with the models and understand the potential impact of changes in our environment.

3.1 Importing a resource

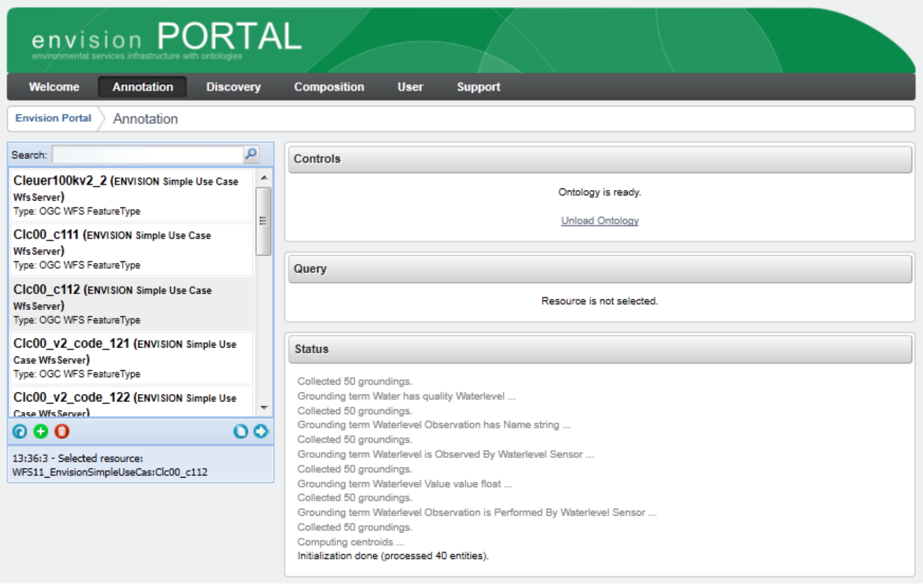
In this example we assume that we know the location of the SOS and the WFS. The Resource Manager Portlet allows for directly importing Web services and Ontologies into the user's personal collection. After selecting the service type and version, the importing procedure will request the service capabilities and store a resource descriptor in the user collection. This service itself cannot be

[10] See `http://ode.apache.org/`

[11] See `http://bpt.hpi.uni-potsdam.de/Oryx/`

[12] Screencasts are available through the project Web site at `http://www.envision-project.eu`

used yet for the annotation and composition. The annotation procedure requires RDF-based service models which can be extended with the references pointing to the ontologies. The composition relies on WSDL service descriptions due to the implementation of the workflows in BPEL. Before moving to the next step, we select the ”Translate Resource” action in the resource portlet. This triggers three subsequent actions: translation, cross-linking, and proxying. The first results in a set of WSDL and RDF files for each feature type for the WFS, observed property for the SOS, or process for the WPS. Cross-linking injects references to the other representations in each of these files and the original capabilities. Many of the components in the ENVISION architecture require access to the other representations: the semantic-aware optimization of the BPEL compositions has for example only access to the WSDL files, but requires the RDF-based service models to reason on the services. Proxying will register the original capabilities file to the semantic proxy, which ensures that the semantic enhancements are also available to standard OGC clients [17]. The following Figure 2 shows the Annotation Page, which contains the Resource Manager Portlet as well as the annotation portlets.



Cleuer100kv2_2 (ENVISION Simple Use Case Wfs Server)
Type: OGC WFS FeatureType

Clc00_c111 (ENVISION Simple Use Case Wfs Server)
Type: OGC WFS FeatureType

Clc00_c112 (ENVISION Simple Use Case Wfs Server)
Type: OGC WFS FeatureType

Clc00_v2_code_121 (ENVISION Simple Use Case Wfs Server)
Type: OGC WFS FeatureType

Clc00_v2_code_122 (ENVISION Simple Use Case Wfs Server)

13:36:3 - Selected resource:
WFS11_EnvisionSimpleUseCas:Clc00_c112

Ontology is ready.

Unload Ontology

Resource is not selected.

```
Collected 50 groundings.
Grounding term Water has quality Waterlevel ...
Collected 50 groundings.
Grounding term Waterlevel Observation has Name string ...
Collected 50 groundings.
Grounding term Waterlevel is Observed By Waterlevel Sensor ...
Collected 50 groundings.
Grounding term Waterlevel Value value float ...
Collected 50 groundings.
Grounding term Waterlevel Observation is Performed By Waterlevel Sensor ...
Collected 50 groundings.
Computing centroids ...
Initialization done (processed 40 entities).
```

Fig. 2. The Annotation Page with the Resource Portlet

3.2 Annotating a Web service

In the next step, we semantically annotate the RDF-based service models resulting from the translation. The annotation procedure itself does not put any constraints on the ontologies (besides being encoded in RDF). The ontologies are grounded, which means each concept and relation is automatically enriched with a rich set of textual documents to enable natural language queries. In this example, we are going to annotate the feature type `clc00_c112` of the WFS. According to the CORINE documentation, this code represents urbanized areas in Europe. After the ontology has been grounded through the "Prepare Ontology" action in the Resource Manager Portlet, the ontology is loaded into the Annotation Portlet. The user inputs the natural language query "populated areas" to find the appropriate concept in the ontology. Due to the grounding, the Query Portlet is able to infer that the concept `UrbanAreas` matches the user query. The user selects the most relevant result, which results in the next step in an annotation linking the feature type `clc00_c112` to the domain concept `UrbanAreas` through a rule injected into the service description.

3.3 Discovering a Web service

A WFS and an SOS have been imported in the personal collection, but a Web service capable of processing and reclassifying the input features according the real-time observations is still missing and unknown to the user. The discovery tool in the ENVISION portal allows for searching for Web services registered to an OGC-compliant catalogue. The catalogue itself might be configured to federate queries to other catalogues, making it possible to cover a potentially very large set of relevant Web services. Discovery of geospatial data traditionally requires the specification of standard keyword based queries as well as the spatiotemporal constraints (e.g., a set of geographic coordinates representing the region the data should cover, and the time interval the data should be valid for). In the case of a processing service only the keyword-based query is useful. The user enters the search term "drought", which results in a list of Web services which have been computed to be relevant for this query. He selects the WPS doing the classification, and adds it to the personal collection visualized in the resource portlet.

3.4 Composing an environmental model

With the composition portlet, the modeler is able to compose services by specifying the control and data flow between the services. The compositions are specified using the BPMN graphical notation used for representing processes. The composition portlet wraps the open source Web-based BPMN Oryx editor. It has been extended with support for including OGC services as part of process compositions and a module that generates executable BPEL code that can be deployed and executed in a BPEL engine. In combination with the Resource Manager Portlet, the composition portlet offers an integrated modeling environment where the user selects services from the resource module, imports them as tasks in the composition, specifies the necessary control and data flow in BPMN, and generates executable BPEL code from the BPMN composition. The following Figure 3 provides a screenshot of the resource portlet and the composition portlet. The composition portlet depicts the composition of an SOS service (GetElevation) and an WFS service (GetWaterLevel) together with their data and control flows. Once the composition is finished, it is transformed into an executable BPEL composition which is stored in the user collection. As final step, the user deploys the composition to the runtime infrastructure via the Deployment Service.

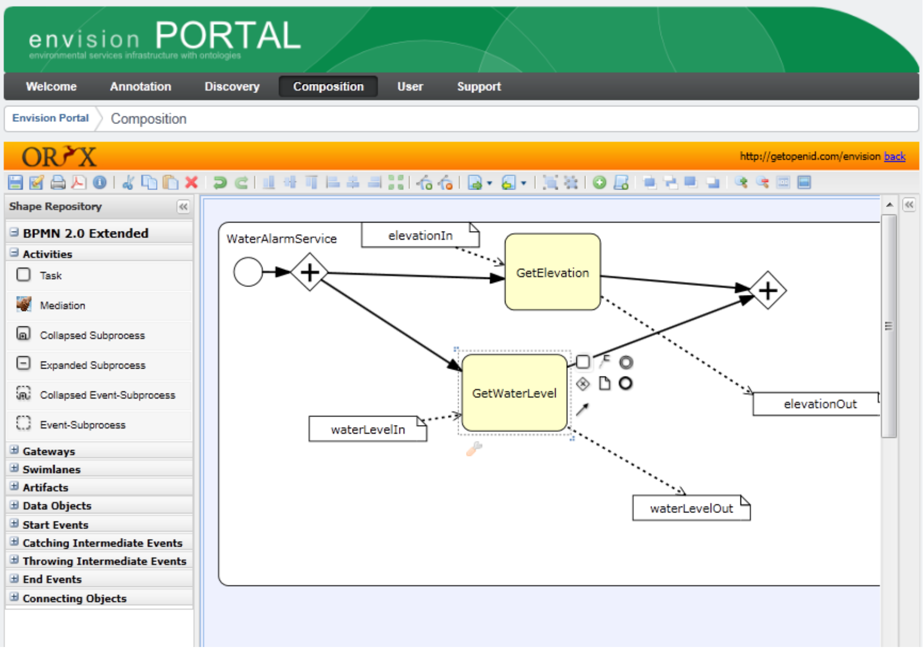


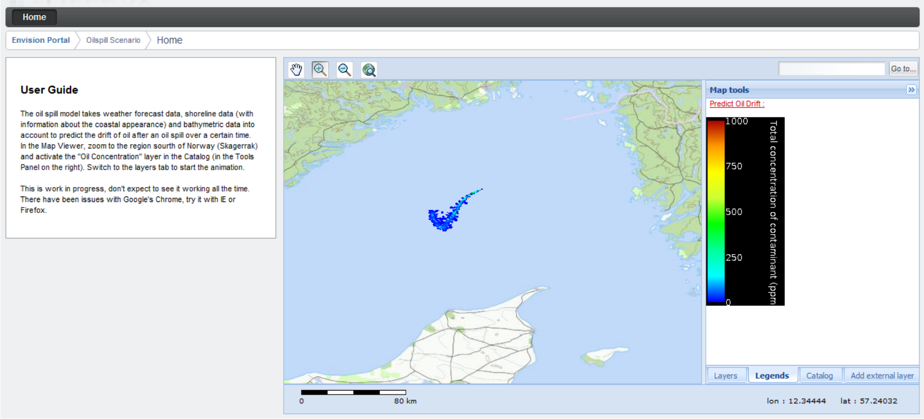
Fig. 3. The Composition Page with the Oryx Workflow Editor

3.5 Creating the Scenario Web site

At this stage, a new environmental model has been created and successfully deployed to the runtime infrastructure. The deployment results in a new Web service endpoint, which, if called, returns the result of the model. In the next step, we are setting up a new Web site with the help of the Portal software which is configured to show this result. The functionality required for this task is built into the Portal Container. The user creates a new sub-portal, defines the Portal Pages, and adds the Portlets to the Portal Pages. Some of the Portlets require configuration (e.g. the locations of the Spatial Data Services have to defined for the MapViewer Portlet). Once finished, the user defines who has access to the Scenario Web site (e.g. is authentication required, or is it open to the public).

3.6 Accessing the environmental model

The Scenario Web site has been published and consists of the MapViewer Portlet showing the results of a composition computing the risk of water shortage for certain regions in France. The MapViewer Portlet has been configured to show the regions exceeding a certain risk of drought, as well as the groundwater sensors. The MapViewer Portlet supports basic map interaction modes such as panning and zooming. Besides that the user is able to toggle layers, display a legend, and select the objects presented on the map. Selecting one of the groundwater sensors displays a timeline of the groundwater observations in the ChartViewer Portlet. The OpenLayer-based MapViewer Portlet also supports the animation of spatial data with a temporal dimension (using the time extension of the OGC WMS). The following Figure 4 shows the result of an oil spill drift model within the MapViewer Portlet.



User Guide

The oil spill model takes weather forecast data, shoreline data (with information about the coastal appearance) and bathymetric data into account to predict the drift of oil after an oil spill over a certain time. In the Map Viewer, zoom to the region sourth of Norway (Skagerrak) and activate the "Oil Concentration" layer in the Catalog (in the Tools Panel on the right). Switch to the layers tab to start the animation.

This is work in progress, don't expect to see it working all the time. There have been issues with Google's Chrome, try it with IE or Firefox.

Fig. 4. A Scenario Web site with an Oil Spill Model in the MapViewer.

4 Open Challenges

In the following section we highlight some of the open research challenges which came up during the first iteration of the implementation, and which will be further investigated in the following years.

Interacting with environmental models on the Web Environmental models are complex software systems whose initial core has been implemented decades ago, and are since then gradually improved and extended. The underlying implementation is based on sometimes ancient software architectures realized with programming languages such as Fortran or Cobolt. This makes it challenging to split up these models into individual reusable software components which can then be migrated to Web services. In addition, invoking models requires rich user interaction. In the oil spill scenario, for example, the domain experts need to be able to constantly adapt the input parameters to calibrate the models. To observe the results, sophisticated data anlyzing tools such as computing and visualizing the cross profile of a submerged oil slick are applied. Making Web-enabled rich user interface components which are generic enough to support different models is an issue which requires a thourough analysis of the end user requirements.

Standardized Web service interfaces for environmental models The environmental models are transformed either as a whole into Web services or split up into individual and reusable components. These componenets are later coupled again through Web service compositions. In the end, the new Web services have to be made accessible to generic spatial data clients to enable the integration into existing geospatial workflows. Web service standards from the W3C or the OGC (e.g. the Web Processing Service) ensure flexibility regarding the supported operations, but also impair reusability for clients expecting well-defined interfaces. Supporting both - reusability and flexibility - requires a compromise which reflects the best of both worlds.

Integration of generic Web services and domain-specific file formats Even though OGC standards have found wide-spread adaption within some communities, they are rarely seen on a global scale. Talking about Web service usually implies the use of WSDL service descriptions and the SOAP standards for encoding the messages. In contrary to OGC services, these Web service don't have predefined semantics for the operations and the parameters. Finding ways to semantically annotate generic Web services with the methodology described above will be further investigated in the next year. But in the environmental community, even the concept of a generic Web service itself is not common. Here, much data is made available as files, often in unique formats. Making this data available to the BPEL workflows requires the implementation of proxy services which translate the Web service requests and return the appropriate files.

Design-time mediation for the composition Support for the specification of mappings between data structures is a necessity when one needs to specify data passing in compositions. Being able to map between the output of a task and the input of another task is part of the actual specification of the composition and necessary for the generation of executable code for the composition. Mappings both at the syntactic and semantic layers are needed, as well as techniques to semi-automatically suggest possible mappings to the user.

Efficiency of runtime mediation The mappings required for linking Web services can range from simply assignments to complex rules including arithmetic or spatial transformations. In the case of environmental data, the number of data entities which have to be processed during the mediation can significantly slow down the execution process. Caching results and the specification of the mediation rules in low-level languages are currently investigated to ensure performant execution.

Inspecting the executed compositions Severe trust issues still impair the uptake of Web services by environmental modeling experts. The Web services are considered to be black boxes, which produce results hard to understand without knowing the internal processes of the model [5]. The Scenario Web sites will include Portlets allowing for inspecting the provenance of the models, by retracing the individual steps in the workflows and inspecting the intermediate results.

Integration of existing ontologies Ontologies come in different flavors. Their applicability for annotating the environmental services depends on the chosen ontology language, the invested effort for its engineering, and the authors' experience in the domain. The technical integration of remote ontologies requires a translation into formats which can be processed by the reasoning engines deployed in ENVISION.

Ontology Maintenance Ontologies usually represent one particular perspective on a domain and have been engineered with one particular application in mind. They hardly ever suit the needs of other domain-specific applications. Integrating existing ontologies relies on techniques to adapt the ontologies for other applications while staying consistent with the original source. Users of the ENVISION Portal will be able to import ontologies into their personal collections. The annotation interface will give them means to dynamically add new concepts or rename existing concepts.

Creating new ontologies If no suitable ontologies exist, the users have to be able to simply create new ontologies without having to know how to write the actual ontology code. A simple concept suggestion Portlet has been created in the first year, which supports the user through data mining techniques to come up with new ontologies. In the next step, users will align these concepts to a foundational ontology by answering automatically generated questions. The new ontologies are published through the Resource Management Portlet, which enables other users to reuse them for other tasks.

This list only includes practical challenges which have to be addressed to move towards the vision of the Model Web. It is obviously far from complete, other research issues such a handling the uncertainty of environmental data or incorporating the real-time aspects of sensor data are also planned to be investigated.

5 Conclusion

This paper presented the ENVISION Portal, a Web-based infrastructure for the discovery, annotation, and composition of environmental services. We have presented a brief discussion of the underlying architecture, and listed a subset of the user interface components developed in the first iteration of the implementation. The implementation of the Portal is part of the research conducted in the ENVISION project, which is scheduled to end in 2012. The open challenges identified in this paper are part of the project's research agenda.

The targeted user community of the ENVISION Portal are scientists and environmental modeling experts willing to share their models to the public. The Portal supports the publication of Scenario Web sites, which are domain-specific portals making the models accessible to users lacking knowledge about the inner working of environmental computer models. We don't claim that the presented infrastructure will be able to cover all possible environmental models. In fact, probably only very few will be ever made available as Web services. The inherent complexity of model algorithms, the sophisticated requirements for the user interfaces, and open questions regarding trust and licensing rights are and will be a major obstacle for the distribution of models to the public. Nonetheless we believe that the ENVISION Portal is one important step towards the vision of shared environmental information systems.

6 Acknowledgements

The presented research has been funded by the European project *ENVISION* (FP7-249120).

References

1. Granell, C. and Díaz, L. and Gould, M., Service-oriented applications for environmental models: Reusable geospatial services, Environmental Modelling & Software 25 (2) (2010) 182–198. doi:10.1016/j.envsoft.2009.08.005.
2. Geller, G. N. and Melton, F., Looking Forward: Applying an Ecological Model Web to assess impacts of climate change, Biodiversity 9 (3&4).
3. Christian, E., Planning for the Global Earth Observation System of Systems (GEOSS), Space Policy 21 (2) (2005) 105–109.
4. Roman, D. and Schade, S. and Berre, A. J. and Bodsberg, N. R. and Langlois, J., Model as a Service (MaaS), 2009.
5. Urvois, M. and Berre, A. J., D1.3 - Experience Report, SWING Project Deliverable (July 2009).
6. M. Andrei, A. Berre, L. Costa, P. Duchesne, D. Fitzner, S. Schade, N. Steinmetz, F. Tertre, L. Vasiliu, SWING : A Geospatial Semantic Web Service Environment, in: Proceedings of AGILE 2008 Workshop "Semantic Web meets Geospatial Applications", 2008.
7. S. Heppner, JSR 286: Java Portlet Specification 2.0 (2008). URL `http://www.jcp.org/en/jsr/detail?id=286`
8. Cox, S., Observations and Measurements - Part 1 - Observation schema (OGC 07-022r1) (Dezember 2007).
9. M. Williams, D. Cornford, L. Bastin, B. Ingram, UncertML : an XML schema for exchanging uncertainty ., in: Proceedings of GISRUK, Manchester, UK, Vol. 44, 2008, pp. 0–3.
10. D. Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley Professional, 2002.

11. H. Michels, P. Maué, Semantics for notifying events in the affecting environment, in: Proceedings of Enviroinfo 2010, Bonn, Germany, 2010.
12. D. Nuescheler, JSR 283: Content Repository for JavaTM Technology API Version 2.0 (2009).
    URL `http://jcp.org/en/jsr/detail?id=283`
13. Diane, J. and Evdemon, J., Web Services Business Process Execution Language Version 2.0, OASIS Standard (April 2007).
14. T. Allweyer, BPMN - Business Process Modeling Notation (German Edition), BoD, 2009.
15. G. Decker, H. Overdick, M. Weske, Oryx An Open Modeling Platform for the BPM Community, in: M. Dumas, M. Reichert, M. C. Shan (Eds.), Proceedings of the 6th Int'l Conference on Business Process Management (BPM 2008), Vol. 5240 of LNCS, Springer Verlag, Milano, Italy, 2008, pp. 382–385–385.
16. Tsalgatidou, A. and Athanasopoulos, G. and Pantazoglou, M., Interoperability Among Heterogeneous Services: The Case of Integration of P2P Services with Web Services., International Journal of Web Service Research 5 (4) (2008) 79–110.
17. P. Maué, H. Michels, M. Roth, Injecting semantic annotations into (geospatial) Web service descriptions, Accepted for Semantic Web Journal (SWJ).