Integrating Web-based Sensor Information into Geospatial Mass-market Applications through OGC Web Processing Services

Theodor Foerster
International Institute for Geo-information Science and Earth Observation (ITC), Enschede, the Netherlands
foerster@itc.nl

Bastian Schaeffer
Institute for Geoinformatics, Muenster, Germany
schaeffer@uni-muenster.de

Simon Jirka
Institute for Geoinformatics, Muenster, Germany
jirka@uni-muenster.de

Johannes Brauner
Professorship of Geoinformation Systems, Technische Universität Dresden, Dresden, Germany
johannes.brauner@tu-dresden.de

Abstract

Sensor data are currently increasingly available on the web through Sensor Web technology. Sensors are an important asset in crisis management situations. However, to make decisions sensor information is required. This information is extracted using geoprocesses provided by for instance Web Processing Services. This paper describes an approach to provide sensor information to ordinary users by integrating sensor data and geoprocesses into mass-market applications. The applicability of the approach is demonstrated by a risk management scenario. The software presented has been developed within the Geoprocessing Community of the 52°North initiative and is available through an Open Source license.

Keywords: Geospatial mass-market applications, OGC WPS, Google Earth, Sensor Web.

1. Introduction

Sharing and accessing web-based geo-information (GI) is a key aspect in geospatial mass-market applications (such as Google Earth™ and Yahoo Maps™) and helps people answer spatially related questions. Currently, data services are organized in Spatial Data Infrastructures (SDIs) and allow anybody to access data on the web from anywhere at anytime. Those data services have matured within the last years, such as the suite of services published by the Sensor Web Enablement (SWE) initiative. However, sensor data available through the so-called Sensor Web often requires certain processing to answer a specific question. As most of the current data are available through web services, the processing should also be established on the web. For this reason, the Open Geospatial Consortium (OGC) released the Web Processing Service (WPS) interface specification [1] and realizes thereby a shift from services providing data towards services providing information. Additionally, Web Processing Services are promising as the availability of network (gigabit bandwidth) and processing capabilities (such as provided by multiple processor cores and advanced processing hardware) increases. In general, Web Processing Services provide a means to customize data offered by data services mostly located in SDIs.

The extraction of information from such web-based sensor data and its integration into current geospatial mass-market applications is promising to provide user-friendly access to up-to-date information and thereby helps users to answer their questions regarding a geospatial context. However, such integration has not been realized yet.

This paper analyzes the technological requirements of geospatial mass-market applications towards integrating sensor information by the means of web-based geoprocesses. Furthermore, it describes an approach to realize this integration. Finally, the paper demonstrates the applicability of the proposed approach by a fire threat use case. This use case demonstrates how ordinary users can access the most current information through their geospatial mass-market application and take actions accordingly. The described scenario addresses a forest fire assessment use case, which is one of four key areas of the OSIRIS project[1].

OSIRIS (Open architecture for Smart and Interoperable networks in Risk management based on In-situ Sensors) is a European integrated project within the Sixth Framework Programme, aligned with GMES (Global Monitoring for Environment and Security). The main goal of the project comprises of the design, development and testing of a service-oriented architecture for risk monitoring and crisis management. A special focus is put on web services for integrating in-situ sensors and sensor data as well as higher level user services. Furthermore, during the design of the OSIRIS architecture, open standards, especially those provided by the OGC, are used as a valuable input.

The tools implementing the presented approach have been developed within the Geoprocessing Community[2] of the 52°North initiative and are available through an Open Source license (GNU General Public License 2).

The remainder of the paper is structured as follows. First the key concepts applied in the presented approach will be described. Section 3 will present the benefits of providing sensor information by the means of web-based geoprocesses. Based on this, Section 4 will present the developed approach. The details about the implementation of the approach is described in Section 5. The developed approach then is applied to the risk management scenario in Section 6. The paper ends with a conclusion and elaborates on future work items.

This paper goes beyond the originally published work of [2] by reviewing work in the context of Sensor Web and providing insights into integrating Sensor Web and web-based geoprocessing.

2. Background

The term *geospatial mass-market application* is closely related to what has been called *Neogeography* [3,4] and *Volunteered Geographic Information* (VGI; [5]). Both concepts describe processes for creating, sharing and annotating geodata (e.g. locations) through web-based applications by the public and can be summarized under the term *Geoweb* [3]. There are several different software vendors active within this market, providing data, applications and services such as Google, Yahoo, Microsoft or NASA. One of the commonly used formats to exchange geospatial-related content within geospatial mass-market applications is *KML*. The following subsections will shortly introduce the KML standard, the WPS interface specification and the components of the Sensor Web.

2.1. The OGC KML standard

KML is widely used for geospatial mass-market applications such as Google Maps and Google Earth. Most lately, it became an official OGC standard [6]. KML is unique in the family of OGC data encodings, as it combines data encoding, styling and the special network facilities, which are called *NetworkLinks* and are also known as dynamic KML. These NetworkLinks are especially interesting in the context of web-based information, as they allow the dynamic integration of remote resources. Therefore, the content of a KML file might become dynamic and provide temporal data (e.g. from sensors). As NetworkLinks use URLs, KML is not only bound to file-based access, but can link any web service, as long as it operates via HTTP-GET and serves KML. NetworkLinks provide additional properties such as update, scheduling etc.

It has to be noted that NetworkLinks have already been integrated in GIS applications such as described by [7], but not for processing purposes.

2.2. OGC Web Processing Service

The WPS interface specification (OGC 2007) describes a standardized set of operations to publish and execute any type of geoprocess on the web. According to the WPS interface specification, a process is defined as an algorithm, calculation or model that operates on spatially referenced data.

In detail, the WPS specification describes three operations, which are all handled in a stateless manner: *GetCapabilities*, *DescribeProcess* and *Execute*. GetCapabilities is common to any type of OGC Web Service and returns service metadata. In case of a WPS it also returns a brief description of the processes offered by the specific WPS instance. To get further information about the hosted processes, the WPS is able to return the process metadata through the *DescribeProcess* operation. This operation provides the description of all parameters, which are required to run the process. Based on this information the client can perform the *Execute* operation upon the designated process. As any OGC Web Service, the WPS communicates through HTTP-GET and HTTP-POST using a messaging based on an OGC-specific XML-encoding.

[1] OSIRIS project website: www.osiris-fp6.eu.

[2] 52°North Geoprocessing Community website: www.52north.org/wps.

Besides that, the WPS interface specification describes mechanisms for asynchronous processing, processing of URL-referenced data and storing of process results. Especially the storing of process results is promising in the context of the presented approach. It allows to access process results server-side whenever required using a simple URL without re-initiating the process itself. Additionally, it is possible to request process results in a raw-data mode.

WPS implementations have already been successfully applied in several projects ranging from groundwater vulnerability analysis [8], bomb threat detection scenarios [9] and map generalization [10]. Additionally, an extensive discussion about the applicability of the WPS and its current drawbacks can be found in [11].

2.3 The Sensor Web

The activities of the OGC are centered on the idea of the so-called Geospatial Web [12]. This describes the aim to integrate geospatial data sources as well as processing capabilities in to the WWW. One aspect of the Geospatial Web, the integration of sensors and (potentially real time) data delivered by these sensors, is addressed by the Sensor Web Enablement Working Group [13].

The SWE Working Group has created a comprehensive framework of standards that aim at fulfilling the following objectives of the Sensor Web:

- Accessing sensor data (observations made by sensors)
- Tasking sensors (e.g. setting the parameters of a sensor)
- Alerting if user-defined alert conditions are matched by sensor measurements (e.g. alert if the water level at a certain location reaches a critical threshold)
- Accessing sensor parameters (e.g. information about the sensor configuration)
- Retrieving descriptions of sensors (sensor metadata)
- Discovering sensors and observations made by sensors.

The resulting framework of standards is divided into two parts (Figure 1): The SWE Information Model addresses the encoding of sensor data and metadata whereas the SWE Service Model consists of standards defining (web service) interfaces that offer the required functionality. In the remaining paragraphs of this subsection a short introduction into the SWE framework will be given to illustrate the architecture.

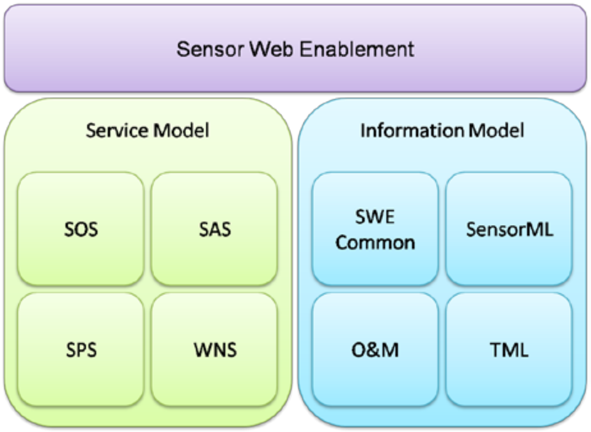

Figure 1: Overview of the SWE framework.

As previously explained the SWE Information Model comprises a set of standards defining the SWE data formats. These encodings provide means for exchanging sensor observations and metadata about the sensors in a well defined way. In detail, the following four standards are part of the SWE Information Model:

- SWE Common [14]: There are several basic building blocks (e.g. simple data types) that are used across the different SWE encodings. These elementary types are standardized within SWE Common.
- Observations and Measurements (O&M) [15,16]: O&M is the encoding for the data captured by sensors (= observations and measurements).
- Sensor Model Language (SensorML) [14]: Besides the data delivered by sensors it is also important to describe metadata about the sensor so that measurements can be interpreted, their quality can be assessed and previous processing steps can be documented. SensorML offers an encoding to provide this kind of metadata in a standardized way.
- Transducer Markup Language (TML) [17]: TML is capable of encoding sensor data as well as metadata. However it was developed as a further standard to provide a data format that is optimized to support data streaming.

Besides defining the data formats for exchanging sensor data and metadata there must also be a common approach is required for interfaces providing sensor related functionality. Such a set of interface standards is defined by the SWE Service Model. Four different standards belong to the SWE Service Model:

- Sensor Observation Service (SOS) [18]: The core functionality of the SOS is providing access to sensor data and metadata. It allows defining filter criteria to select the data on interest but also operations for inserting sensors and observations. Usually SOS instances return sensor data encoded in O&M and sensor metadata encoded in SensorML.
- Sensor Alert Service (SAS) [19]: Whereas the SOS is relying on a pull-based communication model, the SAS is able to push data to subscribers. These subscribers are able to define alert conditions in which they want to receive notifications.
- Sensor Planning Service (SPS) [20]: The SPS covers the issue of tasking sensors. This means that the SPS can be used for submitting tasks to sensors but also to manage these jobs (e.g. deleting, updating, and canceling).
- Web Notification Service (WNS) [21]: Finally, the WNS adds the capability of asynchronous communication to the SWE architecture. Thus, it can be seen as a helper service that can be used by the other SWE service (e.g. by the SAS for transmitting alerts).

3. Sensor Web and Web-based Geoprocesses

The integration of Sensor Web data and web-based geoprocesses (published as WPS) has been described for the application of smoke forecast models [22]. In this study, the authors distinguish between data processing services and data analysis services. Data processing services are considered to be placed in-situ on the actual sensor data service to reduce the amount of transferred data (tight coupling). The data analysis service combines data from distributed sources to create the desired information (loose coupling).

To integrate Sensor Web and web-based geoprocesses successfully, interoperability is crucial. The Sensor Web as described in Section 2.3, with its Information Model creates the foundation to encode the gathered data. The Service Model provides a means to deliver the encoded data in a standardized fashion. Both models build the starting point to transform the data into information.

As already discussed by [22], data processing can either be performed tightly coupled to the sensor (i.e. as a data processing service), or performed in a distributed fashion (data analysis service).

By tightly coupling the sensor with processing capabilities, such as in a simple case a temperature sensor only transmitting average values, the sensor device is able to perform the necessary calculations in-situ. This can be useful in several scenarios, especially for simple processes and low volumes of data. When it comes to processing large volumes of data and complex analysis functions of multiple data, the application of loosely-coupled and distributed services is more useful, since the processing power and storage power does not need to be located where the sensor is. Additionally, such web service-based analysis is reusable in different context with different sensor data. Limited power supply is also in many cases a crucial issue for sensor networks. Thus it is beneficial to process the data outside the sensor network and thereby to decrease the power consumption on the sensor web. Moreover, processing the data in a distributed fashion as a data analysis service turns into a requirement, when accessing multiple data sources is necessary.

By using WPS to generate information from Sensor Web data, there are basically two ways of communication:

- Data sent by value
- Data sent by reference.

Especially transmitting data by reference allows a WPS to obtain the latest available data. This is especially interesting to set up persistent service chains for analyzing latest sensor data. The service chain is set up once and performed whenever it is requested, analyzing latest data (as specified by the reference). Furthermore, the references can be used to implement caching strategies, as the WPS can identify data based on the reference without retrieving it again. However, there is always the problem of retrieving latest data and processing out-dated data (without knowing). Applying a specific caching strategy mostly depends on the actual application scenario. The main aim is to reduce the latency of the system for the end user [23].

When integrating Sensor Web and web-based geoprocesses to analyze real time sensor data the Sensor Observation Service seems to the suitable entry point to access the Sensor Web. The data provided by SOS is encoded as O&M and can be used easily within WPS via reference.

The following section describes the integration of these two services using a mass-market application. It also illustrates the idea of designing a service chain once and running it multiple times based on the latest sensor data.

4. Integrating Sensor Information into Geospatial Mass-market Applications

To integrate web-based sensor information, several requirements of the geospatial mass-market applications have to be met. The major requirement is that the communication pattern (REST architecture & KML encoding) of the geospatial mass-market applications does not have to be changed. Thereby, the sensor information becomes capable of being seamlessly integrated into such applications. This requirement is met by the WPS interface specification, as it allows the invocation of processes via HTTP-GET. Additionally, the WPS interface specification does not foresee any data encoding for its input and output parameters, thus KML is a valid format for the WPS. Finally, as the WPS is able to return process results as raw data without any WPS-specific message overhead it is highly applicable for the integration into geospatial mass-market applications.

The integration of sensor information into mass-market applications is possible through WPS interface due to the following aspects:

- It transforms sensor data into sensor information through geoprocesses
- It converts internally the sensor data from O&M into KML.

As the configuration of such processes is highly complex and not supported by current geospatial mass-market applications, this paper proposes a two-fold approach (Figure 2). At first, the selection and configuration of the process should be done by an expert user, utilizing an expert user interface, most-likely a Geographic Information System (GIS) such as the *User-friendly Desktop Internet GIS* (uDig[3]). At second, the user can integrate the pre-configured process into his geospatial mass-market application of choice. This is possible, as the WPS interface meets the requirements of geospatial mass-market applications, as explained in the previous paragraph. It is important to note, that the pre-configuration is not only necessary because of the lack of support for process configuration in geospatial mass-market applications, but is also highly applicable, as process configuration involves a lot of expert user knowledge. Thus the pre-configuration eases the integration of such information (extracted by geoprocesses) in mass-market applications for the user and is thereby not considered as a drawback.

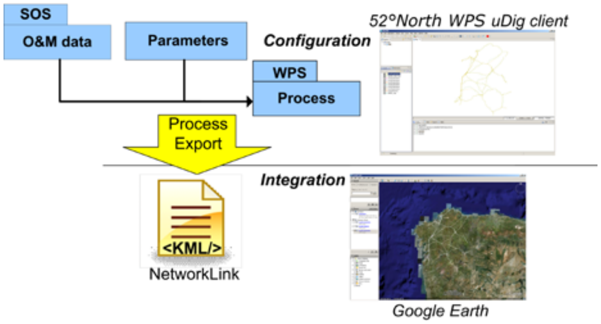

Figure 2: Approach to integrate sensor information in mass-market applications such as Google Earth.

For this study, the processes are configured through the 52°North WPS uDig client. This client application is extensively described in [24]. uDig has been enhanced to export these configured processes as KML, to integrate these geoprocesses and their results into geospatial mass-market applications. The export of the process from uDig to KML can be configured in two ways:

1. Export the KML file as a link to a stored process result. This is the *static* option, in which no process will be triggered when visualizing the KML file in Google Earth. This uses the store functionality of the WPS interface specification
2. Export the KML file as a link, which triggers the process on the WPS. This is the *dynamic* option and enables to trigger the process live, when incorporating the file in Google Earth. This allows one also to set a refresh rate to initiate the process on the server again. It is important to note, that in this case, the WPS process is triggered and if any WPS input data is defined as reference, the (updated) data is fetched and used as the basis for the calculation. This approach allows the processing of the latest available sensor data and thus visualizing the latest results in mainstream applications.

In both cases the files incorporate the links using the *NetworkLink* functionality of KML. Listing 1 shows the generated NetworkLink using the dynamic option in the KML export of uDig (option 2). The generated KML includes an Execute request via HTTP-GET to a spatial Buffer algorithm, which is also used in the scenario described in Section 6. The request references remote O&M data served by a SOS instance.

[3] uDig website: udig.refractions.net.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Folder>
     <name>Buffered Features</name>
             <visibility>0</visibility>
             <open>0</open>
             <description>WPS Layer</description>
               <NetworkLink>
                    <name>WPS Layer</name>
                    …
                    <description>WPS Layer</description>
                    <refreshVisibility>0</refreshVisibility>
                   <Link>
<href>http://geoserver:8080/wps/WebProcessingService?request=ex
ecute&amp;service=WPS&amp;version=1.0.0&amp;Identifier=org.
n52.wps.server.algorithm.SimpleBufferAlgorithm&amp;DataInputs
=FEATURES=@mimeType=text/xml@href= http%3a%2f%2fv-
wupper%2f52nSOSv2_ArcSDE%2fsos%3fService%3dSOS%26Ver
sion%3d0.0.0%26Offering%3dwv.offering2%26observedProperty%
3dwv.phenomenon10%26responseFormat%3dtext%2fxml%3bsubty
pe%3d%2522om%2f0.0.0%2522@Schema=http://schemas.opengis.
net/om/1.0.0/om.xsd;DISTANCE=20&amp;RawDataOutput=BUFF
ERED_FEATURES@mimeType=application/vnd.google-
earth.kml%2Bxml@schema=http://www.opengis.net/kml/2.2</href>
             <refreshMode>onInterval</refreshMode>
             <refreshInterval>20</refreshInterval>
   …
</kml>
```

Listing 1: KML NetworkLink with a WPS-Execute request via HTTP-GET. The request references sensor data provided by SOS.

By supporting these two options (dynamic vs. static), the integration is well-scalable and applicable to scenarios requiring dynamic or even static process results. In case of integrating sensor data, dynamic process results might be more applicable to always obtain the latest sensor information.

It is important to note, that the client is able to perform and export single and chained processes. Chained processes are built up by using the output of a process as input for another process. This task of chaining is performed locally at the client side and does not involve any central processing engine. A more sophisticated approach in the context of web-based geoprocessing by the means of the Business Process Execution Language (BPEL) is described in [25].

The overall architecture with its different components (e.g. WPS and SOS) and clients (e.g. uDig and Google Earth) is depicted in Figure 3.

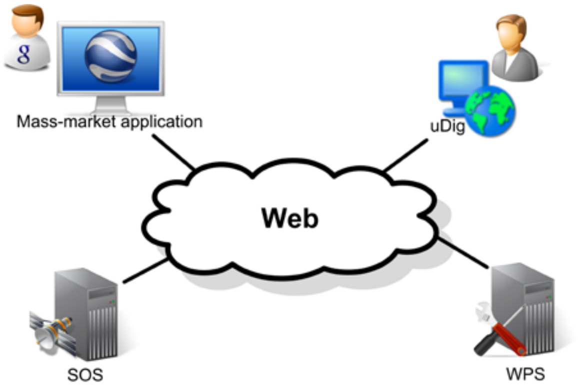

Figure 3: Overview of the involved components.

5. Implementation

This section presents the implementation of the components incorporated in the architecture. The described implementations are available as Open Source software through the 52°North initiative[4]. The implementations described in this section are all based on the Java programming language. The implemented Web Services (WPS and SOS) are deployed in servlet containers such as Tomcat.

5.1 52°North Initiative

As described above the architecture is based on the 52° North WPS and SOS framework.
52° North is an international research and development network which supports the process of advancing early research results to stable implementations that can be productively used in the geospatial domain. To strengthen this process, 52° North is shaped as an open membership initiative that brings together partners from research, industry and practical users.
The implementations published on the 52° North platform are available through a dual license model. This means that all software packages are published under the GNU General Public License 2 (GPL2) and concurrently under a commercial license.

5.2 52°North WPS Framework

The 52°North WPS framework is fully-compliant to the WPS interface specification version 1.0.0 and is developed and maintained by the 52°North Geoprocessing Community. The framework is based on a pluggable architecture, which is extensible regarding the designated processes and data handlers. It can be

[4] 52°North website: www.52north.org.

configured easily through a browser-based user interface and also integrates other geoprocessing functionality such as Sextante [26] and GRASS [27].

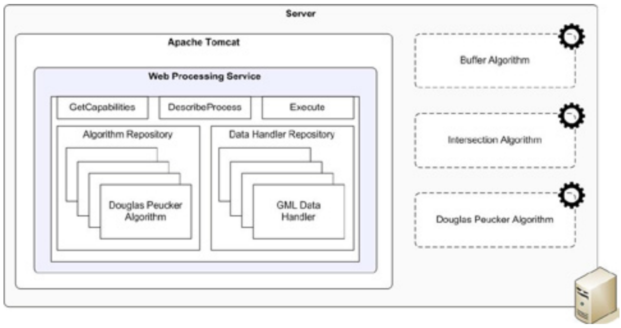

Figure 4: Architecture of 52°North WPS framework.

Besides the features as defined by the WPS interface specification, the framework supports workflow management [28] and grid computing [29]. These aspects are also subject to further research and have been identified as future activities of the Geoprocessing community.

For this study, the WPS has been enhanced to serve the sensor data (encoded as O&M) as KML. In particular, the data models are mapped internally by the WPS. This has been achieved in two steps. At first, the WPS requests the sensor data from the SOS and transforms the resulting O&M data into an internal data model, such as the Geotools feature model. The designated process is then performed on this internal data model. At second, the processed data are transformed into the external output data model, in this case KML. In both cases mappings are required between the O&M and the internal data model (e.g. Geotools feature model) and between the internal data model and KML.

5.3 52°North WPS uDig Client

The 52°North WPS uDig client is designed as a plug-in for uDig [25]. It is able to configure and perform remote functionality provided through the WPS interface. The specific process can be configured through a user-friendly wizard, which is generated on-the-fly, based on the input and output parameters of the designated process. The result of the performed process appears as a separate layer in uDig and can thereby be combined with other resources. As uDig is also able to integrate remote Web Services such as SOS, it is possible to send the data via reference. This allows the WPS to process directly remote resources.

Additionally, the 52°North WPS uDig client supports to export configured WPS processes as KML, which has been developed for the presented study.

5.4 52°North SOS Framework

The 52° North SOS framework is the foundation for this study to serve the sensor data in the presented architecture. It is fully-compliant to the OGC SOS standard version 1.0.0. Besides the mandatory operations of the core profile for accessing sensor data and metadata, it offers full support of the so-called transactional profile. This enables to insert sensors as well as their observations through a standardized web-based interface.

As the 52° North SOS framework relies on a modular design the adaptation and customization of the implementation is supported.

In the presented architecture the 52° North SOS framework serves the sensor data based on a PostgreSQL database (including the PostGIS extension).

6. Use Case Scenario

The scenario is applied to a risk management use case, in which in-situ-sensor data have to be analyzed for assessing a fictive fire threat in Northern Spain. The scenario and the involved services have been extensively presented in [24]. Currently, the services and data are taken from the ORCHESTRA project, which addresses a similar fire risk management scenario [30]. In a later stage of the project, data will be collected by sensors, as it is also the aim of the OSIRIS project. This section will focus on a modification and extension of the OSIRIS fire threat scenario, in which information has to be derived from real-time sensor data and the process results have to be disseminated to inform the public about the current situation.

Other examples of forest fire use cases are presented by [31,32]. For instance [31] present a web-based architecture, which is not based on standards and it is thereby not possible to adopt their architecture and approach to any similar use case. However, the presented approach (Section 4) is applicable to any other use case due to the interchangeable notion of web services. The chosen use case demonstrates the benefits of the approach and the resulting architecture.

According to the approach described in Section 4, the expert configures the process in the 52°North WPS uDig client. In particular the user configures a buffer of the sensor data to indicate potential fire threats and intersecting them with the road data. The buffer operation and the intersection operation are single processes hosted on a WPS. The road data has been additionally simplified to improve the process performance and to improve portrayal at smaller scales. Overall, this allows the expert to assess the parts of the road infrastructure which are at risk by a fire threat. The expert user exports the configured process as a KML file and links it on the national portal site. The citizen (i.e. ordinary user) is now able to visualize the latest analysis results in his/her geospatial mass-market application by loading the KML file from the portal site. He/she can inspect the latest data with underlying base information from areal imagery and/or topography. Thereby, the geospatial mass-market application makes use of distributed Web Services to get the latest information (extracted from sensor data & feature data) for each user request and processes it real-time using OGC WPS. Figure 5 depicts the result of the configured process in uDig and the same process accessed through Google Earth.

7. Conclusion & Outlook

The presented approach enables the seamless integration of sensor information into geospatial mass-market applications by the means of OGC WPS. This is promising, as processes can generate geo-information and especially sensor information, which is required for instance by risk management scenarios. As described in Section 4, the WPS interface specification meets all the requirements for the integration into geospatial mass-market applications. By enabling KML support and the support of the HTTP-GET Execute operation the WPS is capable to be integrated in any geospatial mass-market application. Moreover, as WPS provides the means to extract sensor information from sensor data and to provide this information in KML format, it allows the user to have access to the Sensor Web as such.

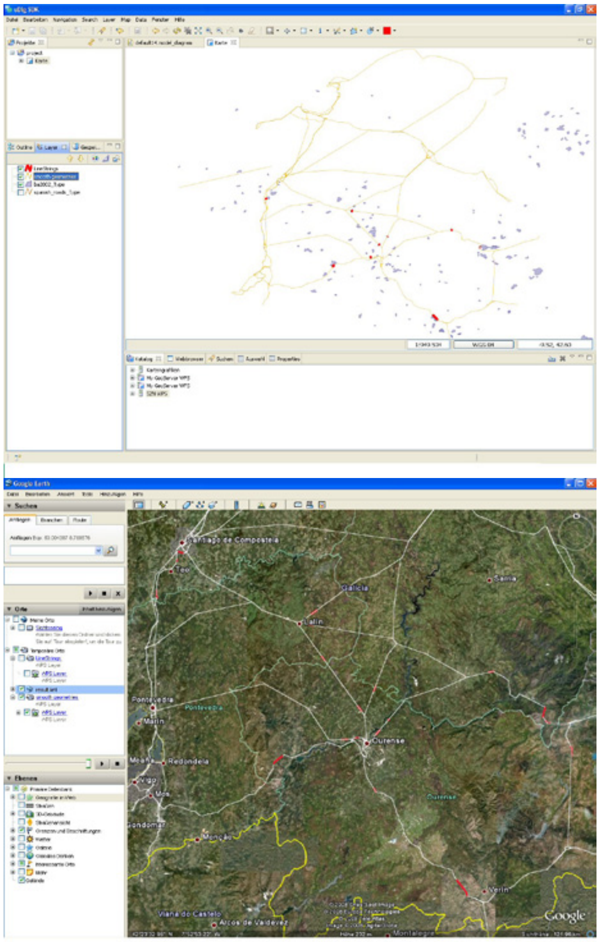

Figure 5: Screenshots of the configured processes in uDig (top) and exported to Google Earth (bottom) – simplified roads & affected road infrastructure (red).

The presented approach is two-fold, as it first allows an expert user to configure the process within an expert client environment (such as uDig). At second, the exported process can be consumed by a geospatial mass-market application such as Google Earth.

The applied use case scenario demonstrates the necessity and applicability of the developed approach for a risk management scenario. Without the integration of sensor information into such applications, citizens would be unaware about current information and could not act accordingly in times of danger. The visualization of sensor information, such as affected road infrastructure combined with static satellite imagery or topography, provides sufficient information to the ordinary user regarding the aimed scenario. Additionally, the approach is interesting for research communities, which need to exchange latest research results in terms of process results (e.g. latest results of climate change models).

The approach is scalable as the sensor information can be integrated as dynamic or static processes (Section 4). It is important to note that the presented approach is fully compliant with the applied standards (KML, WPS, SOS & O&M), without amending or modifying any interfaces or encodings. Overall, the specifications for the OGC WPS interface and family of specifications in the context of the Sensor Web as well as the dynamic KML have shown great flexibility to enrich information on the currently evolving GeoWeb by enabling to integrate sensor information into geospatial mass-market applications.

In particular, the *NetworkLink* feature of KML and the capability of WPS to process referenced data (i.e. data sent by reference), allows the geospatial mass-market applications to integrate service chains accessing the latest data. This is a key aspect in developing architectures for risk management scenarios in the future.

As explained by [33], the performance of web-based geoprocesses as integrated into geospatial mass-market applications is of major interest in the future. Adequate performance is crucial for the usability of the application and opens a new market beyond enterprise applications for the Geospatial Web. Also, the integration of more complex and intelligent geoprocess chains is subject to research.

References

[1] OGC, *OpenGIS Web Processing Service*, Open Geospatial Consortium, 2007.

[2] T. Foerster, B. Schaeffer, J. Brauner, and S. Jirka, “Integrating OGC Web Processing Services into Geospatial Mass-Market Applications,” *International Conference on Advanced Geographic Information Systems & Web Services, 2009.*, Cancun, Mexiko: IEEE, 2009, pp. 98-103.

[3] P. McFedries, “The new geographers,” *IEEE Spectrum*, vol. December 2007, 2007, p. 96.

[4] A. Turner, *Introduction to Neogeography*, O'Reilly, 2006.

[5] M. Goodchild, “Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0,” *International Journal of Spatial Data Infrastructures Research*, vol. 2, 2007, pp. 24-32.

[6] OGC, *OGC KML*, Open Geospatial Consortium, 2008.

[7] T.M. Smith, O.K. Norman, and V. Lakshmanan, “Utilizing Google Earth as a GIS platform for weather applications,” *22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Atlanta, US: 2006.

[8] C. Kiehle, “Business logic for geoprocessing of distributed data,” *Computers & Geosciences*, vol. 32, 2006, pp. 1746-1757.

[9] B. Stollberg and A. Zipf, “OGC Web Processing Service Interface for Web Service Orchestration - Aggregating Geo-processing Services in a Bomb Threat Scenario,” *Proceedings of Web and Wireless Geographical Information Systems*, Heidelberg: Springer-Verlag, 2007, pp. 239-251.

[10] T. Foerster and J.E. Stoter, “Establishing an OGC Web Processing Service for generalization processes,” *ICA workshop on Generalization and Multiple Representation*, Portland, Oregon, USA: 2006.

[11] A. Friis-Christensen, N. Ostlander, M. Lutz, and L. Bernard, “Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions,” *Transactions in GIS*, vol. 11, Dec. 2007, pp. 799-818.

[12] D. Roman and E. Klien, “SWING - A Semantic Framework for Geospatial ServicesRoman,” *The Geospatial Web*, A. Scharl and K. Tochtermann, eds., London, UK: Springer, 2007, pp. 229-234.

[13] M. Botts, G. Percivall, C. Reed, and J. Davidson, *OGC Sensor Web Enablement: Overview And High Level Architecture*, Wayland, MA, USA: OGC, 2007.

[14] M. Botts and A. Robin, *OpenGIS Sensor Model Language (SensorML) Implementation Specification 1.0.0*, Wayland, MA, USA: OGC, 2007.

[15] S. Cox, *Observations and Measurements - Part 1 - Observation schema 1.0.*, Wayland, MA, USA: OGC, 2007.

[16] Cox, *Observations and Measurements - Part 2 - Sampling Features 1.0*, Wayland, MA, USA: OGC, 2008.

[17] S. Havens, *OpenGIS Transducer Markup Language (TML) Implementation Specification 1.0.0*, Wayland, MA, USA: OGC, 2007.

[18] A. Na and M. Priest, *Sensor Observation Service 1.0*, Wayland, MA, USA: OGC, 2006.

[19] I. Simonis, *OGC Sensor Alert Service Candidate Implementation Specification 0.9*, Wayland, MA, USA: OGC, 2006.

[20] I. Simonis and P.C. Dibner, *OpenGIS Sensor Planning Service Implementation Specification 1.0*, Wayland, MA, USA: OGC, 2007.

[21] I. Simonis and J. Echterhoff, *Draft OpenGIS Web Notification Service Implementation Specification 0.0.9*, Wayland, MA, USA: OGC, 2006.

[22] S. Falke, E. Dorner, B. Dunn, and D. Sullivan, “Processing Services in Earth Observation Sensor Web Information Architectures,” *Proceedings of the Earth Science Technology Conference 2008*, College Park, Maryland: NASA, 2008.

[23] S. Sivasubramanian, G. Pierre, M.V. Steen, and G. Alonso, “Analysis of Caching and Replication Strategies for Web Applications,” *IEEE Internet Computing*, vol. 11, 2007, pp. 60-66.

[24] T. Foerster and B. Schaeffer, “A client for distributed geo-processing on the web,” *W2GIS*, G. Tayler and M. Ware, eds., Springer, 2007, pp. 252-263.

[25] B. Schaeffer and T. Foerster, “A Client for Distributed Geo-Processing and Workflow Design,” *Journal for Location Based Services*, vol. 2, Sep. 2008, pp. 194-210.

[26] V. Olaya, “SEXTANTE - the free geoprocessing library,” Notti: University of Nottingham, 2009.

[27] J. Brauner and B. Schaeffer, “Integration of GRASS functionality in web based SDI service chains,” Cape Town, South Africa: 2008, pp. 420-429.

[28] B. Schaeffer, “Towards a Transactional Web Processing Service (WPS-T),” *Proceedings of the 6th Geographic Information Days*, E. Pebesma, M. Bishr, and T. Bartoschek, eds., Muenster, Germany: Institute for Geoinformatics, 2008, pp. 91-116.

[29] B. Baranski, “Grid Computing Enabled Web Processing Service,” *Proceedings of the 6th Geographic Information Days*, E. Pebesma, M. Bishr, and T. Bartoschek, eds., Muenster, Germany: Institute for Geoinformatics, 2008, pp. 243-256.

[30] A. Friis-Christensen, L. Bernard, I. Kanellopoulos, J. Nogueras-Iso, S. Peedell, S. Schade, and C. Thorne, “Building Service Oriented Application on top of a Spatial Data Infrastructure - A Forest Fire Assessment Example,” *Proceedings of the 9th Agile Conference on Geographic Information Science*, Visegrad, Hungary: 2006, pp. 119-127.

[31] S. Yassemi and S. Dragicevic, “Web Cellular Automata: A Forest Fire Modeling Approach and Prototype Tool,” *Cartography and Geographic Information Science*, vol. 35, 2008, pp. 102-115.

[32] J. Black, C. Arrowsmith, M. Black, and W. Cartwright, “Comparison of Techniques for Visualising Fire Behaviour,” *Trans*, vol. 11, 2007, pp. 621-635.

[33] J. Brauner, T. Foerster, B. Schaeffer, and B. Baranski, “Towards a Research Agenda for Geoprocessing Services,” *12th AGILE International Conference on Geographic Information Science*, J. Haunert, B. Kieler, and J. Milde, eds., Hanover, Germany: IKG, Leibniz University of Hanover, 2009.