

Open Geospatial Consortium, Inc.

Date: 2008-09-12

Reference number of this document: OGC 08-058r1

Version: 0.1.0

Category: [Discussion Paper](#)

Editor: Stefan Falke

OGC[®] OWS-5 Earth Observation Web Processing Services (WPS) Engineering Report

[Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.](#)
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Preface

This draft document serves to describe the use of web processing services and the OGC Web Processing Service (WPS) in earth observation (EO) applications. It provides an overview of web processing services and a description of developments related to earth observation implementations of OGC WPS in the OGC OWS-5 testbed.

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

Contents		Page
1	Introduction	1
1.1	Scope	1
1.2	Document contributor contact points	1
1.3	Revision history.....	1
1.4	Future work.....	2
2	References	2
3	Terms and definitions	2
4	Conventions.....	2
4.1	Abbreviated terms	2
5	Earth Observation Processing through Web Services	3
6	OGC Processing Service Standards.....	3
6.1	OGC Web Processing Service (WPS).....	3
6.2	The Web Coverage Processing Service (WCPS).....	4
6.3	Web Image Classification Service (WICS)	4
6.4	OGC Testbed Experiences (OWS-3, OWS-4, OWS-5).....	5
6.5	Available WPS Software	5
7	SOAP/WSDL for Processing Services	5
7.1	SOAP/WSDL Background	6
7.2	Using SOAP/WSDL as an Interface to Web Processing Services.....	8
8	REST for Processing Services.....	8
9	Other Distributed Processing Frameworks	8
9.1	ESRI Geoprocessing Service	8
9.2	The Earth Science Modeling Framework (ESMF)	8
9.3	Kepler for Scientific Processing Workflows	9
10	OWS-5 EO WPS Implementations.....	9
10.1	Nearest Neighbor Analysis	9
10.1.1	Nearest Neighbor Analysis Service through SOAP/WSDL.....	9
10.2	EO-1 Smoke Classifier	10
10.3	OWS-5 EO WPSes within context of larger use cases.....	13
	Annex A.....	14
	Annex B.....	25
	Bibliography	26

Figures		Page
----------------	--	-------------

Figure 1. EO-1 Smoke algorithm as an OGC Web Processing Service10
Figure 2. EO-1 Smoke algorithm as SOAP/WSDL Service11
Figure 3. The smoke classifier using the DataFed workflow engine as the grid processing platform.....12
Figure 4. Example Execution of EO-1 Smoke Classifier OGC WPS.....12
Figure 5. Multiple uses of smoke forecasts and products.....13

OGC® OWS-5 Earth Observation Web Processing Services (WPS) Engineering Report

1 Introduction

1.1 Scope

This document reviews the use of web processing services within the context of earth observation data processing and analysis. It also describes OWS-5 activities in developing, implementing and testing the OGC Web Processing Service standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Stefan Falke	Northrop Grumman Corporation

1.3 Revision history

Date	Release	Editors	Primary clauses modified	Description
07 Dec 07	0.0.1	S. Falke, E. Dorner, B. Dunn		Draft document
22 Mar 07	0.0.2	S. Falke	3, 4, 5, 10	
31 Mar 07	0.0.3	S. Falke, B. Dunn	3, 10, Annex A, Annex B	Final OWS-5

1.4 Future work

It is anticipated that this document will be updated as the EO Fire and Smoke Sensor Web scenario and demonstrations evolve and its experiences with web processing service implementations expands.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS[®] Web Services Common Specification*

OGC 06-187r1, *OWS-4 Workflow IPR: Workflow descriptions and lessons learned*

OGC 06-035r1, *Web Coverage Processing Service (WCPS)*

OGC 05-017, *Web Image Classification Service (WICS) Implementation Specification*

OGC 05-007r7, *Web Processing Service*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

3.1

Earth observation

the gathering of measurements about earth features and phenomena using any type of sensor

3.2

Web processing service

a web service that in some way filters, fuses, or aggregates data (spatial-temporal data in the case of this Engineering Report)

4 Conventions

4.1 Abbreviated terms

EO Earth Observations

WPS Web Processing Service

5 Earth Observation Processing through Web Services

Earth observations consist of data collected by satellite, surface and airborne sensors. The data are of multiple types including gridded and point coverages. Processing of these data occur at multiple points across the sensor observation-to-knowledge information flow, including on-board sensor processing, data re-formatting processing, and data analysis processing. Conducting the data flow through web services allows components to be distributed and coupled to form an end-to-end processing chain of services consisting of contributions from diverse and distributed service providers.

In this document, we distinguish between two types of processing services:

- 1) **data processing services** that can be done as part of the data access. Types of processing that can be conducted on the data provider service and often results in a reduction in the amount of data transferred to the client.
- 2) **data analysis services** that include processing that combines data from multiple distributed servers or services to create derived information.

Each type of processing service can be handled differently in processing service chains. In coupling data processing with data access, an objective is to minimize the amount of data being transferred from the data server. The processing is conducted on the data server before data are transferred rather than transferring the entire data set to a separate processing service that executes a data reduction process, such as subsetting or resampling. This type of processing as part of a data access request is described by the OGC Web Coverage Processing Service or OGC Web Feature Service – Transaction.

Data analysis services can involve data from multiple servers, where the data access to each server is coupled in a service workflow chain. The data sources serve as inputs to the analysis service in which they are analyzed according to the processing service analysis algorithm.

This OGC EO WPS Engineering Report focuses on data analysis services.

6 OGC Processing Service Standards

6.1 OGC Web Processing Service (WPS)

The Web Processing Service (WPS) is described in OGC 05-007r7, *Web Processing Service* as a standardized interface that facilitates the publishing of geospatial processes, and the discovery of and binding to those processes by clients. Processes include any algorithm, calculation or model that operates on spatially referenced data. WPS is a generic interface in that it does not identify any specific processes that are supported. WPS can be thought of as an abstract model of a web service, for which profiles need to be developed to support use, and standardized to support interoperability.

The OGC WPS specifies three mandatory operations that can be requested by a client and performed by a server. Those operations are: GetCapabilities, DescribeProcess, and Execute.

1. GetCapabilities returns a Capabilities metadata document that describes the processes offered by the WPS server. The GetCapabilities operations should be implemented using the HTTP GET transfer using KVP encoding.
2. DescribeProcess returns information about a specific process, including inputs and outputs required to execute that process. The DescribeProcess operation should be implemented using HTTP GET transfer using KVP encoding. HTTP POST transfer with XML encoding is optional.
3. Execute runs the specified WPS process. The Execute operations should be implemented using HTTP POST transfer with XML encoding. HTTP GET transfer with KVP encoding is optional.

The OGC WPS Standard does not specify output formats.

The OGC WPS allows standardized profiles to foster interoperability within particular applications. An Application Profile describes how WPS shall be configured to serve a process that is recognized by OGC. An Application Profile is defined to include

1. An OGC URN that uniquely identifies the process (mandatory)
2. A reference response to a DescribeProcess request for that process (mandatory).
3. A human-readable document that describes the process and its implementation (optional, but recommended).
4. A WSDL description for that process (optional).

The application profile could be further refined to address the standard name space used in the OGC WPS. If one considers the OGC WPS standard as specifying the syntax and structure of the GetCapabilities, DescribeProcess and Execute requests, an application profile could foster interoperability by specifying the semantics of those requests. The profile might also specify output formats, such as the netCDF-CF convention used in the atmospheric science community.

6.2 The Web Coverage Processing Service (WCPS)

The OGC WCPS offers a coverage language which allows phrasing of arbitrary complex operations by functional composition during a Web Coverage Service data access request.

The OGC WCPS can be categorized as a data processing service where the processing is done through the same service that conducts the data access.

6.3 Web Image Classification Service (WICS)

The Web Image Classification Service (WICS) Implementation Specification defines interfaces for image supervised and unsupervised classification. It includes interfaces for

GetCapabilities, GetClassification, TrainClassifier, and DescribeClassifier. An evaluation of the WICS for WPS and WPS Application Profiles was not undertaken during OWS-5 but should be included in future revisions of this document.

6.4 OGC Testbed Experiences (OWS-3, OWS-4, OWS-5)

The OGC's Web Services Interoperability Program phase 4 (OWS-4) implemented workflows for solving various problems, including EO related web processing services for creating derived products from satellite data. In OWS-4, a binary WPS was implemented that provided a “map algebra” interface for two gridded datasets. The binary processing service was used in OWS-4 to calculate a flood product from EO-1 satellite imagery. OWS-5 will build upon OGC WPS work in OWS-4 in that it will develop workflows to process observations suitable for decision support. It will investigate and implement various processing methods for using subsets of observation datasets and predictive model outputs. In particular the earth observation scenario will involve the discovery, integration, processing, chaining and visualization of earth observations and model outputs to demonstrate the application of standards in describing, discovering, accessing and tasking satellites and ground-based sensor installations in a sequence of analysis activities that deliver information required by decision makers in response to national, regional or local emergencies.

6.5 Available WPS Software

Software packages that have implemented OGC WPS for standing up geospatial web processing services.

Python Web Processing Service (PyWPS) <http://pywps.wald.intevation.org>

Available under a General Public License (GNU) PyWPS is an OGC WPS implementation written in the Python programming language and has native support for GRASS GIS.

52 North WPS <http://52north.org>

Written in Java the 52 North WPS supports all WPS mandatory operations plus Execute process via HTTP GET. The 52 North implementation has been updated for the OGC WPS 1.0 standard.

7 SOAP/WSDL for Processing Services

It is valuable for comparing SOAP/WSDL is a framework commonly used for developing and executing web processing services. The OGC WPS standard document (OGC 05-007r7) and the OWS 5 SOAP/WSDL Common Engineering Report (OGC 08-009r1) compares and contrasts WPS and WSDL.

7.1 SOAP/WSDL Background

SOAP is a simple messaging framework for exchanging structured and typed information in a decentralized, distributed environment over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics [3]. A SOAP message is formally specified as an XML infoset, which provides an abstract description of its contents.

“The SOAP specification describes four major components: formatting conventions for encapsulating data and routing directions in the form of an envelope, a transport or protocol binding, encoding rules, and a Remote Procedure Call (RPC) mechanism. The envelope defines a convention for describing the contents of a message, which in turn has implications on how it gets processed. A protocol binding provides a generic mechanism for sending a SOAP envelope via a lower-level protocol such as HTTP. Encoding rules provide a convention for mapping various application datatypes into an XML representation. Finally, the RPC mechanism provides a way to represent remote procedure calls and their return values” [2].

SOAP messages consists of an <env:Envelope> element that contains an optional <env:Header> element and a <env:Body> element. SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but SOAP can support many Message Exchange Patterns, including but not limited to, one-way messages, request/response interactions, and request/multiple response conversations [3].

SOAP supports XML-based content exchange and exchange modeled as RPC. The XML-based content exchange model consists of XML-based contents conforming to some application-defined schema exchanged via SOAP messages [3]. The RPC model defines a uniform model for representing a RPC request and response using the SOAP infrastructure [2]. “The fundamental requirements for an RPC call are that the body element contains the method name and the parameters and that the parameters are accessible via accessors. In addition, SOAP has provisions for encoding datatypes, method signature, header data, and the URI that represents the destination” [2].

SOAP’s messaging formats brought about a need to describe operational information in an equally structured way. Web Services Description Language (WSDL) was introduced to address that need.

WSDL is an XML-based language used to describe web services and indicate a way for potential clients to interact with the described services. Machines can determine from the WSDL document what operations are available and how to invoke them without manual pre-configuration between the two. WSDL enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as “how” and “where” that functionality is offered [1]. A complete WSDL definition contains all of the information necessary to invoke a Web service. WSDL describes four critical pieces of data:

1. Interface information describing all publicly available functions
2. Data type information for all message requests and message responses

3. Binding information about the transport protocol to be used (how to access the service)
4. Address information for locating the specified service (where to access the service)

Table 1 lists the main elements of a WSDL 2.0 document.

Table 1. Main elements of a WSDL document.

Element Name	Description
Types	Defines the kinds of messages that the service will send and receive; usually defined in XML Schema
Interface	Specifies abstract set of operations supported by one or more endpoints; operations represent a simple interaction between the client and the service
Binding	Specifies concrete message format and transmission protocol details for an interface
Service	Specifies a single interface that the service will support, and a list of endpoint locations where that service can be accessed

WSDL enables a web service to be modeled into two parts: abstract and concrete. The first two elements (types and interface) are abstract definitions of the Web service interface. These elements provide abstract definitions in a platform and language independent manner for the data being exchanged and the operations being performed by a service. For instance, WSDL describes a Web service in terms of the messages it sends and receives through a type system such as XML schema. Each operation specifies the types of messages that the service can send or receive as part of that operation. Each operation also defines a message exchange pattern that indicates the sequence and cardinality of the messages to be transmitted. A total of eight MEPs are defined in WSDL. An interface groups these operations without any commitment to transport or wire format [1].

The last two elements (binding and service) describe the concrete details of how the abstract interface maps to messages on the wire. A binding specifies concrete message format and transmission protocol details for an interface. An endpoint associates a network address with a binding, and finally, a service groups together endpoints that implement a single interface [1].

WSDL defines a SOAP binding extension and a HTTP binding extension. The SOAP binding extension enables web services applications to use SOAP. The HTTP binding extension enables web services applications to use HTTP and HTTPS methods such as GET, POST, PUT, and DELETE [1].

WSDL and SOAP go hand-in-hand. WSDL determines the high-level application protocol. In other words, for each operation, the number of parameters, the type of each parameter, and how those parameters are to be sent through the wire (serialization) are defined through WSDL. On the other hand, SOAP is a low-level protocol for exchanging messages defined by WSDL between services. SOAP by itself cannot determine the high-level application protocol without WSDL.

7.2 Using SOAP/WSDL as an Interface to Web Processing Services

An example of a web processing service that uses SOAP/WSDL is the `Grass_Vector_ValueExtraction` service developed by GMU. This service extracts raster values from a raster map at a set of vector point locations. The body of the SOAP request consists of the following two input string parameters: vector URL and a raster URL. The body of the SOAP response consists of an XML string containing the vector data and the extracted raster value.

8 REST for Processing Services

In the OWS-5 testbed, a REST-ful workflow using OpenWFE was demonstrated that included OGC WPS service for EO-1 smoke classification as described in this document (see Annex B).

9 Other Distributed Processing Frameworks

The review and revision of the OGC WPS standard is strengthened by placing the OGC WPS within the context of distributed processing activities within the earth science community.

9.1 ESRI Geoprocessing Service

The ESRI suite of server-based GIS programs includes Model Builder that allows creation of GIS module workflows that, in turn, can be published as services. It is primarily designed to work across ESRI products but includes a web service interface.

9.2 The Earth Science Modeling Framework (ESMF)

The Earth Science Modeling Framework is a National Science Foundation (NSF) funded initiative to develop a standard framework for coupling atmospheric science model components. While it is not based on web services, it provides a useful reference point for understanding how the atmospheric science user domain conceptually connects distributed processing components. The ESMF architecture is defined in three tiers, 1) an infrastructure of utilities and data structures for creating model components, 2) a superstructure for coupling them, and 3) the individual model components (ala processing services) that are situated between the creation infrastructure and coupling superstructure.

9.3 Kepler for Scientific Processing Workflows

Kepler is an open-source workflow software application for the analysis and modeling of scientific data that provides standard methods and interfaces for creating and describing data flow among discrete analysis and modeling components. Kepler is based on the open-source Ptolemy II visual modeling system. It is a desktop application but supports web services as components of its workflow.

10 OWS-5 EO WPS Implementations

10.1 Nearest Neighbor Analysis

The Nearest Neighbor WPS interface provides an Execute Distance process for calculating the distance between a point from one dataset, and its nearest neighbor in a second dataset. In the test case, the points are observed fire locations from the NOAA Hazard Mapping Service (HMS) reviewed satellite product and the NASA EO1 satellite hot pixel product. The WPS interface requires five parameters to execute the Distance process:

- p1 – name of first dataset, values can be either NOAA or EO1.
- date1 – date of observations for dataset 1 (yyyy-mm-dd).
- p2 – name of second dataset, values can be either NOAA or EO1.
- date2 – date of observations for dataset 2 (yyyy-mm-dd).
- bbox – area of interest (x1,y1,x2,y2)

When the Execute Distance process is called with the appropriate parameters, the WPS will make two OGC WFS requests to obtain the appropriate sets of data (EO1: <http://fire.geobliti.com/wfs> or NOAA: <http://gdsg.ngdc.noaa.gov/firesMapserver/servlet/fireswfs>). Once the data are returned via the WFS requests, the data are inserted into an Oracle database and the Oracle Spatial distance calculation query is performed. The results of the query are then returned to the WPS client as XML.

10.1.1 Nearest Neighbor Analysis Service through SOAP/WSDL

The WPS Nearest Neighbor service was also developed as a SOAP/WSDL service using the Apache Axis2 framework. This was done by first creating a WSDL document describing the service. The WSDL2Java tool was used to generate the skeleton code of the web service. The business logic developed as part of the WPS was inserted into the skeleton. The WSDL2Java tool was executed a second time to generate the client proxy code. Finally, the client was developed, which calls the proxy code.

The Apache Axis2 framework was used for the development of SOAP/WSDL web processing services. Axis2 is a Java-based framework for creating both the client and server parts of a web service. It includes a WSDL2Java tool that generates the web service skeleton code and client proxies from a WSDL document.

10.2 EO-1 Smoke Classifier

During OWS-5, there was a desire to explore the creation of an EO-1 smoke product through the use of a web processing service. A couple of EO-1 smoke delineation algorithms were identified through a literature review. However, it should be noted that the literature highlighted the difficulties in deriving smoke from hyperspectral images and no algorithm has been deemed consistently reliable. We started with algorithms found in literature and set up processing service infrastructure to test and refine algorithms for EO-1. The two smoke classification algorithms for EO-1 imagery implemented in the WPS were:

- Small particles (Fine smoke)
 - Ratio of wavelengths 490nm and 2200nm (Band 14/Band 205)
- Large Particles (Coarse smoke)
 - Ratio of wavelengths 430nm and 510nm (Band 8/Band 16)

The two algorithms were implemented into an OGC WPS 1.0 (see Figure 1). The WPS required three inputs: URL pointer to the “numerator” EO-1 geotiff, URL pointer to the “denominator” EO-1 band geotiff, and the specified output format (e.g., geotiff). The WPS, in turn uses the two EO-1 URLs to formulate requests to the GMU GRASS SOAP web services. The actual computing processing to calculate the band ratios is done on the GMU server and a URL to the resulting geotiff is returned to the WPS. The ratio geotiff is then processed using a classification function on the GMU GRASS server to create a binary geotiff distinguishing smoke from all other non-smoke pixels on the geotiff.

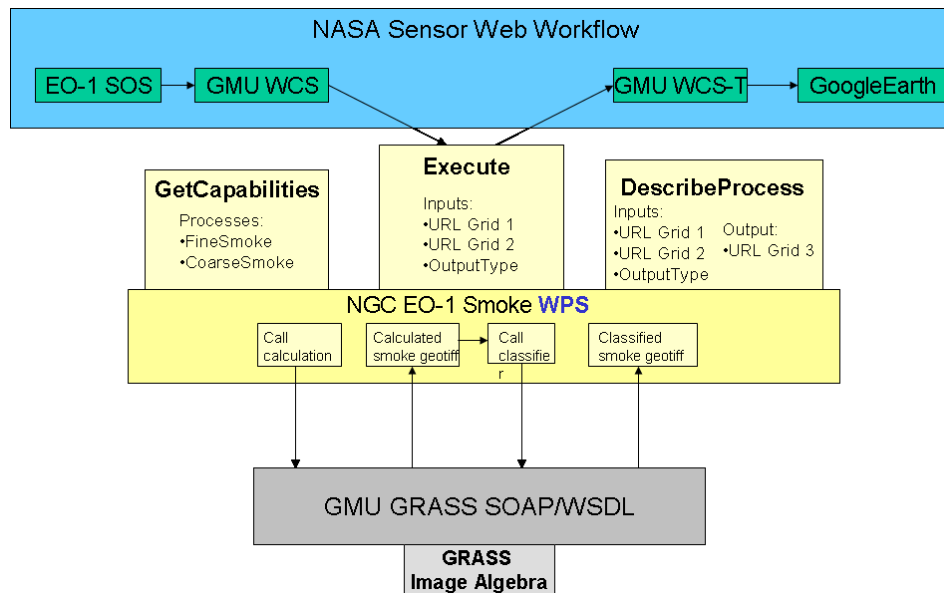


Figure 1. EO-1 Smoke algorithm as an OGC Web Processing Service

The same workflow and smoke classification could be executed with the smoke algorithm wrapped as a SOAP service (Figure 2).

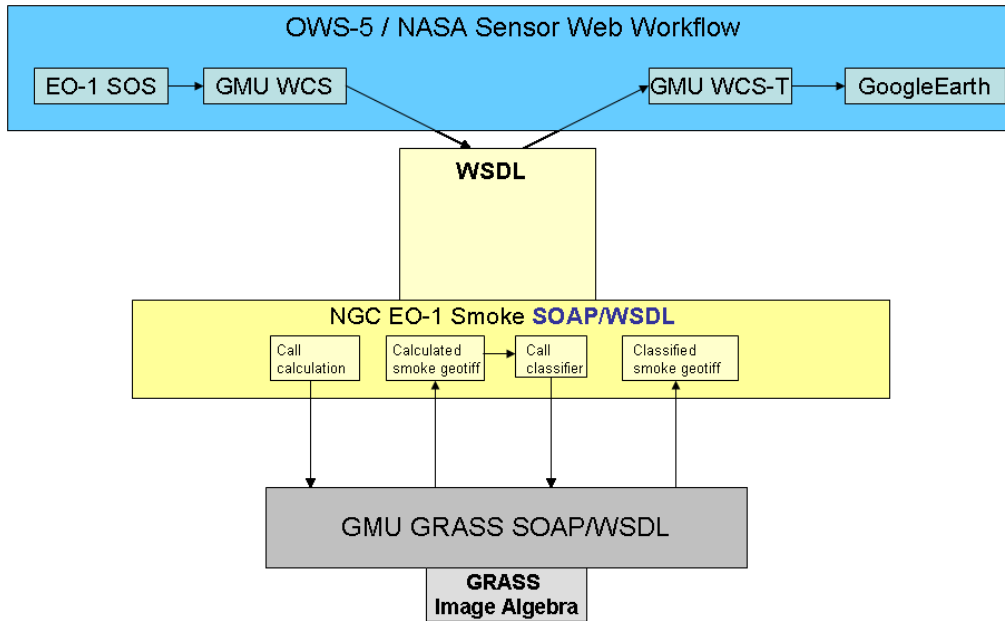


Figure 2. EO-1 Smoke algorithm as SOAP/WSDL Service

Multiple workflow engines are available for executing earth observation web processing services, including BPEL, OpenWfe, Kepler, ESRI Model Builder, etc. Figure 3 illustrates a WPS wrapping of the DataFed workflow engine to execute the smoke classification algorithms. Alternatively, the DataFed URL-based execution request of its workflow could be integrated directly into the overall sensor web workflow, by passing the WPS. The point being that the workflow and execution of the smoke classification algorithms can be configured multiple ways and it raises the question where the standard interfaces should reside.

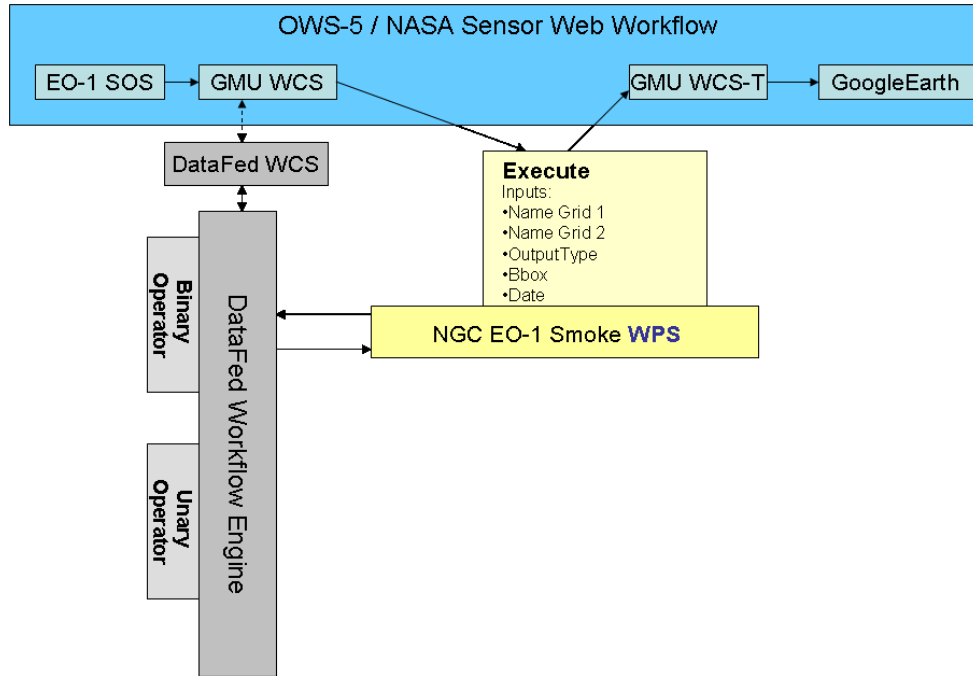


Figure 3. The smoke classifier using the DataFed workflow engine as the grid processing platform.

Figure 4 illustrates the EO-1 smoke classifier OGC WPS using an image captured over Southern California on October 23, 2007 during the large wildfires. The only geotiff that the sensor web workflow “sees” is the final classified image but the intermediate image is shown in the figure for illustration purposes.

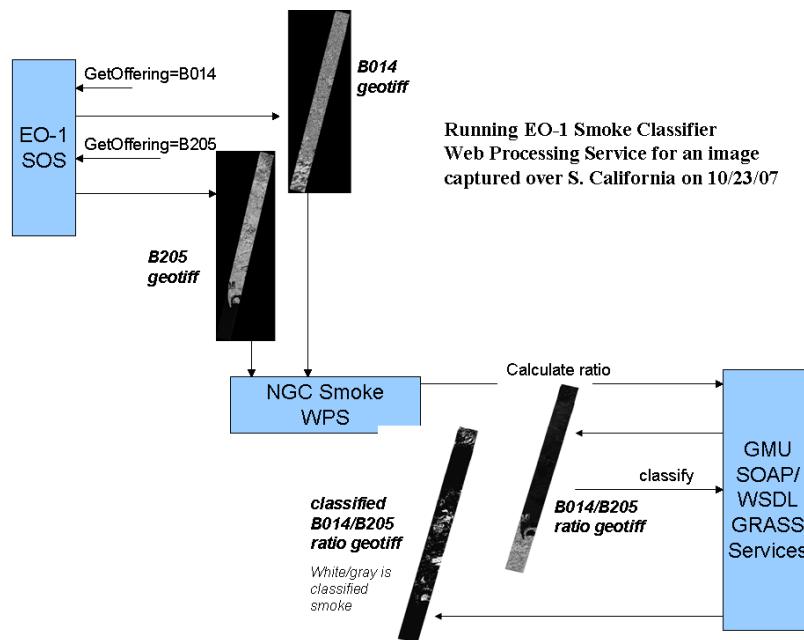


Figure 4. Example Execution of EO-1 Smoke Classifier OGC WPS

A portrayal service could potentially be included in the overall workflow after the OGC WPS to adjust the colorization and other rendering settings before the geotiff is visualized in a web mapping client.

10.3 OWS-5 EO WPSes within context of larger use cases

The two web processing services developed during OWS-5 are subcomponents of an envisioned set of related processing, analysis, and tasking services that includes smoke forecasts driven by sensed fire locations, validated by observation derived smoke products, and used to identify spatial-temporal areas of interest for sensor tasking (Figure 5).

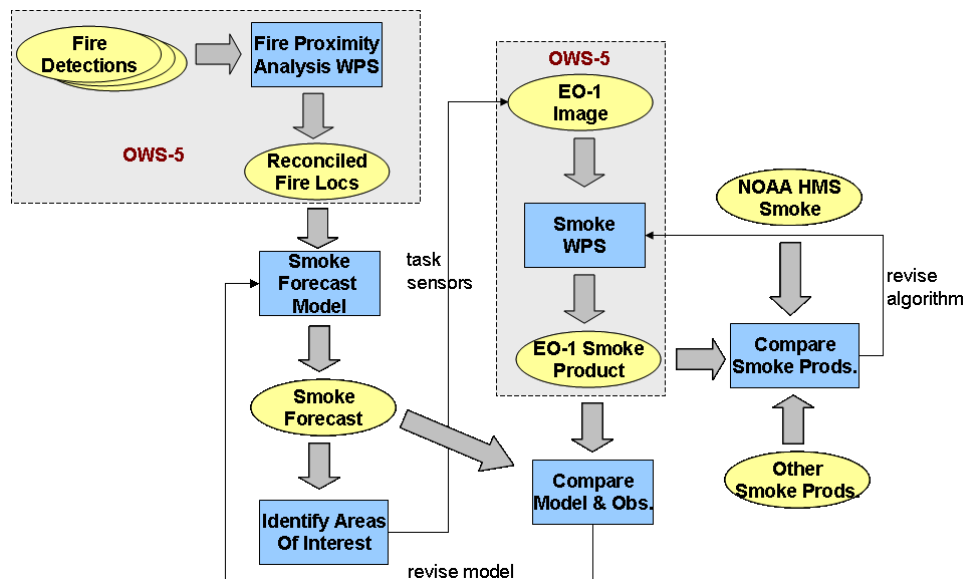


Figure 5. Multiple uses of smoke forecasts and products.

Annex A

Annex A provides example requests and XML results for the OGC WPS implementations for the Fire Proximity Analysis and EO-1 Smoke Classifier services.

Example GetCapabilities Request:

<http://ww1.geoenterpriselab.com/NGCWPS/Wps?service=Wps&version=1.0.0&request=GetCapabilities>

Resulting Capabilities Document:

```
<Capabilities version="0.4.0"
xsi:schemaLocation="http://www.opengespatial.net/wps.\wpsGetCapabilities.xsd">
<ows:ServiceIdentification>
<ows:Title>Northrop Grumman TASC WPS Server</ows:Title>
<ows:Abstract>Northrop Grumman WPS Server</ows:Abstract>
<ows:Keywords>
<ows:Keyword>WPS</ows:Keyword>
<ows:Keyword>geospatial</ows:Keyword>
<ows:Keyword>geoprocessing</ows:Keyword>
</ows:Keywords>
<ows:ServiceType>WPS</ows:ServiceType>
<ows:ServiceTypeVersion>0.2.0</ows:ServiceTypeVersion>
<ows:ServiceTypeVersion>0.1.0</ows:ServiceTypeVersion>
<ows:Fees>NONE</ows:Fees>
<ows:AccessConstraints>NONE</ows:AccessConstraints>
</ows:ServiceIdentification>
<ows:ServiceProvider>
<ows:ProviderName>Northrop Grumman IT-TASC</ows:ProviderName>
<ows:ProviderSite xlink:href="http://ww1.geoenterpriselab.com/" />
<ows:ServiceContact>
<ows:IndividualName>Brian Dunn</ows:IndividualName>
<ows:PositionName />
<ows:ContactInfo>
<ows:Phone>
<ows:Voice>+1 314 259-7843</ows:Voice>
</ows:Phone>
<ows:Address>
<ows:DeliveryPoint>1010 Market Street #1740</ows:DeliveryPoint>
<ows:City>Saint Louios</ows:City>
<ows:AdministrativeArea>MO</ows:AdministrativeArea>
<ows:PostalCode>63101</ows:PostalCode>
<ows:Country>USA</ows:Country>
```

```

<ows:ElectronicMailAddress>brian.dunn@ngc.com</ows:ElectronicMailAddress>
</ows:Address>
</ows:ContactInfo>
</ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata>
<ows:Operation name="GetCapabilities">
<ows:DCP>
<ows:HTTP>
<ows:Get xlink:href="http://ww1.geoenterpriselab.com/NGCWPS/Wps?"/>
</ows:HTTP>
</ows:DCP>
</ows:Operation>
<ows:Operation name="DescribeProcess">
<ows:DCP>
<ows:HTTP>
<ows:Get xlink:href="http://ww1.geoenterpriselab.com/NGCWPS/Wps?"/>
</ows:HTTP>
</ows:DCP>
</ows:Operation>
<ows:Operation name="Execute">
<ows:DCP>
<ows:HTTP>
<ows:Get xlink:href="http://ww1.geoenterpriselab.com/NGCWPS/Wps?"/>
</ows:HTTP>
</ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
<ProcessOfferings>
<Process processVersion="1.0">
<ows:Identifier>FineSmoke</ows:Identifier>
<ows:Title>FineSmoke</ows:Title>
<ows:Abstract>
The FineSmoke process distinguishes small smoke particles based on the algorithm of
Band14 / Band205 (490nm/2200nm)
</ows:Abstract>
</Process>
</ProcessOfferings>
</Capabilities>

```

Example DescribeProcess Request:

<http://ww1.geoenterpriselab.com/NGCWPS/Wps?service=Wps&version=1.0.0&request=DescribeProcess&Identifier=FineSmoke>

Resulting XML:

```

<ProcessDescriptions
xsi:schemaLocation="http://www.opengeospatial.net/wps.\wpsDescribeProcess.xsd">
<ProcessDescription processVersion="2" storeSupported="false"
statusSupported="false">
<ows:Identifier>FineSmoke</ows:Identifier>
<ows:Title>FineSmoke</ows:Title>
<ows:Abstract>
The FineSmoke process distinguishes small smoke particles based on the algorithm of
Band14 / Band205 (490nm/2200nm)
</ows:Abstract>
<ows:Metadata xlink:title="spatial"/>
<ows:Metadata xlink:title="geometry"/>
<ows:Metadata xlink:title="distance"/>
<ows:Metadata xlink:title="GML"/>
<DataInputs>
<Input>
<ows:Identifier>host1</ows:Identifier>
<ows:Title>host1</ows:Title>
<ows:Abstract>
host name for location of first image (ww1.geoenterpriselab.com)
</ows:Abstract>
<MinimumOccurs>1</MinimumOccurs>
</Input>
<Input>
<ows:Identifier>path1</ows:Identifier>
<ows:Title>path1</ows:Title>
<ows:Abstract>
path location for first image (/EO1/FirstImage.tif)
</ows:Abstract>
<MinimumOccurs>1</MinimumOccurs>
</Input>
<Input>
<ows:Identifier>host2</ows:Identifier>
<ows:Title>host2</ows:Title>
<ows:Abstract>
host name for location of second image (ww1.geoenterpriselab.com)
</ows:Abstract>
<MinimumOccurs>1</MinimumOccurs>
</Input>
<Input>
<ows:Identifier>path2</ows:Identifier>
<ows:Title>path2</ows:Title>
<ows:Abstract>
path location for first image (/EO1/SencondImage.tif)
</ows:Abstract>
<MinimumOccurs>1</MinimumOccurs>
</Input>

```

```

</DataInputs>
<ProcessOutputs>
<Output>
<ows:Identifier>results</ows:Identifier>
<ows:Title>result</ows:Title>
<ows:Abstract>Returns xml with a url to the resulting tif image</ows:Abstract>
<ComplexOutput defaultFormat="text/XML" defaultEncoding="base64"
defaultSchema="">
<SupportedComplexData>
<Format>text/XML</Format>
<Encoding>UTF-8</Encoding>
<Schema/>
</SupportedComplexData>
</ComplexOutput>
</Output>
</ProcessOutputs>
</ProcessDescription>
</ProcessDescriptions>

```

Example Execute Request:

```

http://ww1.geoenterpriselab.com/NGCWPS/Wps?service=Wps&version=1.0.0&request=
execute&Identifier=FineSmoke&DataInputs=band014=http://eo1.geobliki.com/sos?REQ
UEST=GetObservation%26OFFERING=EO1H0410362007329110PF%26OBSERVED
PROPERTY=B014%26FORMAT=Dimage/geotiff;band205=http://eo1.geobliki.com/sos
?REQUEST=GetObservation%26OFFERING=EO1H0410362007329110PF%26OBSER
VEDPROPERTY=B205%26FORMAT=Dimage/geotiff

```

Resulting XML:

```

<results><url>
http://65.123.203.154:8099/grassweb/map_calculation_20080331_0934_1961532.tif</url>
</results>

```

Fire Distance WPS

Example GetCapabilities Request:

```

http://ww1.geoenterpriselab.comt/NgcWps/Wps?Service=wps&Request=GetCapabilities
&Version=0.0.4

```

Resulting Capabilities Document:

```

<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns:xlink="http://www.w3.org/1999/xlink" version="0.4.0"
xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wps
..\wpsGetCapabilities.xsd">
  <ows:ServiceIdentification>
    <ows:Title>Northrop Grumman TASC WPS Server</ows:Title>
    <ows:Abstract>Northrop Grumman WPS Server Developed for NASA SAMITS
project</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>WPS</ows:Keyword>
      <ows:Keyword>geospatial</ows:Keyword>
      <ows:Keyword>geoprocessing</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>0.2.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>0.1.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>Northrop Grumman IT-TASC</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://ww1.geoenterpriselab.com/"/>
    <ows:ServiceContact>
      <ows:IndividualName>Brian Dunn</ows:IndividualName>
      <ows:PositionName>Research Software Engineer</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>+1 314 259-7843</ows:Voice>
        </ows:Phone>
        <ows:Address>
          <ows:DeliveryPoint>1010 Market Street
#1740</ows:DeliveryPoint>
          <ows:City>Saint Louios</ows:City>
          <ows:AdministrativeArea>MO</ows:AdministrativeArea>
          <ows:PostalCode>63101</ows:PostalCode>
          <ows:Country>USA</ows:Country>

        <ows:ElectronicMailAddress>brian.dunn@ngc.com</ows:ElectronicMailAddress>
      </ows:Address>
    </ows:ContactInfo>
  </ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get
xlink:href="http://ww1.geoenterpriselab.com/NgcWps/Wps?"/>

```

```

        </ows:HTTP>
    </ows:DCP>
</ows:Operation>
<ows:Operation name="DescribeProcess">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get
xlink:href="http://ww1.geoenterpriselab.com/NgcWps/Wps?"/>
        </ows:HTTP>
    </ows:DCP>
</ows:Operation>
<ows:Operation name="Execute">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get
xlink:href="http://ww1.geoenterpriselab.com/NgcWps/Wps?"/>
        </ows:HTTP>
    </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
<ProcessOfferings>
    <Process processVersion="1.0">
        <ows:Identifier>Distance</ows:Identifier>
        <ows:Title>Distance</ows:Title>
        <ows:Abstract>Returns the distance in KM between two observed points
from NOAA and/or
EO1 datasets</ows:Abstract>
    </Process>
</ProcessOfferings>
</Capabilities>

```

Example Describe Process Request:

<http://ww1.geoenterpriselab.comt/NgcWps/Wps?Service=wps&Request=DescribeProcesses&Identifier=Distance&Version=0.0.4>

Resulting XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessDescriptions xmlns="http://www.opengeospatial.net/wps"
xmlns:wps="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wps
..\wpsDescribeProcess.xsd">
    <ProcessDescription processVersion="2" storeSupported="false"
statusSupported="false">
        <ows:Identifier></ows:Identifier>

```

```

<ows:Title></ows:Title>
<ows:Abstract></ows:Abstract>
<ows:Metadata xlink:title="spatial" />
<ows:Metadata xlink:title="geometry" />
<ows:Metadata xlink:title="distance" />
<ows:Metadata xlink:title="GML" />
<DataInputs>
  <Input>
    <ows:Identifier>p1</ows:Identifier>
    <ows:Title>Data Source 1</ows:Title>
    <ows:Abstract>Webservice to call for first source of
data</ows:Abstract>
    <LiteralData>
      <ows:AllowedValues>NOAA,
EO1</ows:AllowedValues>
    </LiteralData>
    <MinimumOccurs>1</MinimumOccurs>
  </Input>
  <Input>
    <ows:Identifier>p2</ows:Identifier>
    <ows:Title>Data Source 2</ows:Title>
    <ows:Abstract>Webservice to call for second source of
data</ows:Abstract>
    <LiteralData>
      <ows:AllowedValues>NOAA,
EO1</ows:AllowedValues>
    </LiteralData>
    <MinimumOccurs>1</MinimumOccurs>
  </Input>
  <Input>
    <ows:Identifier>date1</ows:Identifier>
    <ows:Title>Observation Date One</ows:Title>
    <ows:Abstract>Observation date used to call data from Data
Source 1</ows:Abstract>
    <LiteralData>
      <Date format="YYYY-MM-DD"/>
    </LiteralData>
    <MinimumOccurs>1</MinimumOccurs>
  </Input>
  <Input>
    <ows:Identifier>date2</ows:Identifier>
    <ows:Title>Observation Date Two</ows:Title>
    <ows:Abstract>Observation date used to call data from Data
Source 2</ows:Abstract>
    <LiteralData>
      <Date format="YYYY-MM-DD"/>
    </LiteralData>

```



```

        <MinimumOccurs>1</MinimumOccurs>
    </Input>
    <Input>
        <ows:Identifier>bbox</ows:Identifier>
        <ows:Title>Observation Bounding Box</ows:Title>
        <ows:Abstract>Observation area of interest </ows:Abstract>
        <BoundingBoxData CRS=""/>
        <MinimumOccurs>1</MinimumOccurs>
    </Input>
</DataInputs>
<ProcessOutputs>
    <Output>
        <ows:Identifier></ows:Identifier>
        <ows:Title></ows:Title>
        <ows:Abstract>GML</ows:Abstract>
        <ComplexOutput defaultFormat="text/XML"
defaultEncoding="base64" defaultSchema=" ">
            <SupportedComplexData>
                <Format>text/XML</Format>
                <Encoding>UTF-8</Encoding>
                <Schema></Schema>
            </SupportedComplexData>
        </ComplexOutput>
    </Output>
</ProcessOutputs>
</ProcessDescription>
</ProcessDescriptions>

```

Example Execute Request:

<http://ww1.geoenterpriselab.com/NgcWps/Wps?Service=wps&Request=Execute&Identifier=Distance&DataInputs=p1,NOAA,p2,NOAA,date1,2007-10-26,date2,2007-10-27,bbox,-120.146484375,32.10118973232094,-114.609375,35.817813158696616&Version=0.0.4>

Resulting XML:

```

<FireDistanceCalculation>
<items>
<item recordCount="1">
<gml:Point obstime="2007-10-29 04:35:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.923,33.391</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 18:45:00.0" satellite="MODIS TERRA">
<gml:coordinates>-116.923,33.391</gml:coordinates>
</gml:Point>
<distance units="KM">0.0</distance>
</item>
<item recordCount="2">

```

```

<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.576,33.738</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.576,33.738</gml:coordinates>
</gml:Point>
<distance units="KM">0.0</distance>
</item>
<item recordCount="3">
<gml:Point obstime="2007-10-29 04:35:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.897,33.375</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 09:20:00.0" satellite="MODIS AQUA">
<gml:coordinates>-116.897,33.375</gml:coordinates>
</gml:Point>
<distance units="KM">0.0</distance>
</item>
<item recordCount="4">
<gml:Point obstime="2007-10-29 04:35:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.919,33.372</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 04:57:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.919,33.372</gml:coordinates>
</gml:Point>
<distance units="KM">0.0</distance>
</item>
<item recordCount="5">
<gml:Point obstime="2007-10-29 04:35:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.902,33.404</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.913,33.403</gml:coordinates>
</gml:Point>
<distance units="KM">1.0292739</distance>
</item>
<item recordCount="6">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.912,33.393</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 18:45:00.0" satellite="MODIS TERRA">
<gml:coordinates>-116.923,33.391</gml:coordinates>
</gml:Point>
<distance units="KM">1.0471793</distance>
</item>
<item recordCount="7">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.564,33.739</gml:coordinates>

```

```

</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.576,33.738</gml:coordinates>
</gml:Point>
<distance units="KM">1.117521</distance>
</item>
<item recordCount="8">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.574,33.728</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-117.576,33.738</gml:coordinates>
</gml:Point>
<distance units="KM">1.124555</distance>
</item>
<item recordCount="9">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.925,33.401</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 18:45:00.0" satellite="MODIS TERRA">
<gml:coordinates>-116.923,33.391</gml:coordinates>
</gml:Point>
<distance units="KM">1.1246144</distance>
</item>
<item recordCount="10">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.9,33.394</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 18:45:00.0" satellite="MODIS TERRA">
<gml:coordinates>-116.91,33.383</gml:coordinates>
</gml:Point>
<distance units="KM">1.5343187</distance>
</item>
<item recordCount="11">
<gml:Point obstime="2007-10-29 23:30:00.0" satellite="GOES-WEST">
<gml:coordinates>-116.927,33.411</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.913,33.403</gml:coordinates>
</gml:Point>
<distance units="KM">1.5758454</distance>
</item>
<item recordCount="12">
<gml:Point obstime="2007-10-29 10:22:00.0" satellite="AVHRR NOAA-18">
<gml:coordinates>-116.938,33.409</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 20:16:00.0" satellite="AVHRR NOAA-18">

```

```
<gml:coordinates>-116.913,33.403</gml:coordinates>
</gml:Point>
<distance units="KM">2.4189115</distance>
</item>
<item recordCount="13">
<gml:Point obstime="2007-10-29 04:35:00.0" satellite="AVHRR NOAA-17">
<gml:coordinates>-116.875,33.388</gml:coordinates>
</gml:Point>
<gml:Point obstime="2007-10-28 09:20:00.0" satellite="MODIS AQUA">
<gml:coordinates>-116.897,33.375</gml:coordinates>
</gml:Point>
<distance units="KM">2.503886</distance>
</item>
</items>
</FireDistanceCalculation>
```

Annex B

The wildfire based scenario serves as the basis for the OWS-5 earth observation workflow. Data sources include EO-1 imagery and derived products, such as fire hot spots and smoke, Unmanned Aerial system imagery and a derived fire location product, and “routine”, daily fire location information collected by satellites (e.g., MODIS) and on-ground observations.

The OWS-5 EO workflow will use the EO-1 smoke classification OGC WPS to test the ability for using OGC WPS to process SOS and WCS accessed imagery to create a EO-1 smoke product. The Earth Observation workflow included OGC Web Processing Services within its OWS-5 demonstration that created multiple Earth Observing workflows using:

- OpenID and OAuth for security
- RESTful interfaces for SWE services
- Workflow definition using OpenWFE, BPEL and SensorML
- SOS and KML for data access and visualization
- SMS and Atom Publication for notifications
- Theme-based, satellite tasking was demonstrated
- Use of web processing services within the context of earth observation data processing and analysis
- Proposed harmonization of SPS with WPS and Workflows
- Workflow discovery

Bibliography

- [1] Booth, D., Liu, C. (eds.): Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, W3C Recommendation (26 June 2007), <http://www.w3.org/TR/wsd120-primer/>
- [2] Chappell, D., Jewell, T.: Java Web Services, O'Reilly & Associates, 2002
- [3] Nilo Metra, E., Lafon, Y. (eds.): SOAP Version 1.2 Part 0: Primer (Second Edition), W3C Recommendation (27 April 2007), <http://www.w3.org/TR/soap12-part0/>
- [4] Michael, C and Ames, D Evaluation of the OGC Web Processing Service for Use in a Client-Side GIS, OSGeo Journal, Vol. 1, May 2007