Processing Services in Earth Observation Sensor Web Information Architectures

Stefan Falke
Northrop Grumman and
Washington University
1010 Market St., Suite 1740
St. Louis, MO 63101 USA

Eric Dorner
Northrop Grumman
1010 Market St., Suite 1740
St. Louis, MO 63101 USA

Brian Dunn
Northrop Grumman
1010 Market St., Suite 1740
St. Louis, MO 63101 USA

Don Sullivan
NASA Ames Research Center
MS 242-4
Moffett Field, CA 94035 USA

Abstract-**Processing of earth observation 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. 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. Data analysis services can involve data from multiple servers, where the data access to each server is coupled in a service workflow chain. In this paper, we examine the development and implementation of standards-based web processing services within service oriented architectures. In particular, we evaluate the Open Geospatial Consortium Web Processing Service (WPS) Standard, web service workflows, and non-web service approaches to coupling processing components for wildfire and smoke data access, analysis and forecast modeling.**

1. INTRODUCTION

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.

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. In this paper, we focus on data analysis services.

Data analysis services also include forecast modeling, such as weather forecasts, that assimilate multiple sources of observation data with predication algorithms to generate a future estimation. Integration of forecasting modeling into service oriented frameworks has be limited to date but new standards and capabilities for conducting distributed processing have set the stage for more integration of model and sensors in service oriented frameworks.

2. WEB PROCESSING SERVICES

Multiple approaches exist for capturing spatial-temporal data processing algorithms as web processing services, including the standard SOAP/WSDL web services, OGC Web Processing Services standard [1], [2]. The present paper focuses on the comparison between SOAP/WSDL and WPS services. A comprehensive discussion that would also address processing services described as workflows, such as through in BPEL, Sensor Model Language, or Workflow Chaining Services (WfCS), is beyond the present paper's scope.

The OGC Web Processing Service (WPS) is described 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 key value pair (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.

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)

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, the Types element describes the format of the message a web service sends and receives through a schema definition language 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 (MEP) 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 Binding and Service elements 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].

In order to execute a SOAP/WSDL service, a SOAP request is generated and sent to the service. A SOAP framework, such as Apache Axis2, can used to generate client proxies which will handle the SOAP requests and responses, thereby avoiding the need to directly deal with them in code.

A SOAP/WSDL request shares common elements with the WPS GetCapabilities, DescribeProcess and Execute requests [3]. While it is not possible to directly create a one-to-one mapping between the structure of WSDL and WPS, Tables 1 and 2 attempt to generally related components from the two standards for describing and invoking processing services.

3. FORECAST MODELS AS WEB PROCESSING SERVICES

Forecast models can be characterized as web processing services; they use various data inputs, process that input data in some way using a set of algorithms, and derive an output product. The Earth Science Modeling Framework (ESMF) provides a modular approach to connecting distributed model components [4]. ESMF is not web service based but does share similar principles with service oriented architectures and serves as proof that shared model components are beneficial to a future forecast modeling systems.

Many of the typical data sources that feed into forecast models are available through standard web service interfaces, including surface meteorological observations. Likewise, the data used to compare and validate model output are available, such as surface measurements and satellite derived products. As service oriented frameworks become adopted for data access and processing, integration of models into the same framework will expose models to a wider user base and will simplify the process of bringing sensor data into model and analysis operations.

Models and data analysis algorithms are beginning to be incorporated into service oriented architectures both as WSDL implementations and WPS implementations [5], [6]. The experiences in implementing models as processing services will help address similarities and differences in standards as, ultimately, the objective is for even the heterogeneous and independent sources of model components to be able to interoperate.

TABLE 1.

SUBSET OF ELEMENTS IN WPS OPERATIONS

WPS Operation	**WPS Element**[1]	**WPS Description**	**Corresponding WSDL Element in which information is described**
GetCapabilities	Operations Metadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests.	Service
	Process Offerings	Unordered list of brief descriptions of the processes offered by the server	Interface, Service
DescribeProcess	Process Description response	Full description of process, including all input and output parameters	Binding
	DataInputs	List of the required and optional inputs to this process	Types
	ProcessOutputs	List of the required and optional outputs from executing this process	Types
Execute	Identifier	Unambiguous identifier or name of a process	Embodied in a SOAP request
	DataInputs	List of inputs provided to this process execution	Embodied in a SOAP request
	Response Form	Defines the response type of the WPS, either raw data or XML document. If absent, the response shall be a response document which includes all outputs encoded in the response.	Embodied in a SOAP request

[1] The WPS elements listed here only represent a subset of the required and optional elements of the respective WPS operations.

TABLE 2.

SUBSET OF ELEMENTS IN WSDL DOCUMENT

Element Name	**Description**	**Corresponding WPS Operation in which information is described**
Types	Defines the format of the messages that the service will send and receive; usually defined in XML Schema	DescribeProcess
Interface	Specifies an abstract set of operations supported by one or more endpoints. Operations represent a simple interaction between the client and the service	GetCapabilities, DescribeProcess
Binding	Specifies concrete message format and transmission protocol details for an interface (how to access a service)	DescribeProcess
Service	Specifies a single interface that the service will support, and a list of endpoint locations where that service can be accessed (where to access a service)	GetCapabilities

4. ACCESSING & PROCESSING FIRE & SMOKE RELATED DATA

We envision the application of service oriented frameworks to wildfire smoke forecasting systems. Fire related applications are well suited for use by distributed web services. A variety of government, academic, commercial, and non-profit web sites disseminate an impressive collection of fire related data. The data available on these http or ftp sites include fire location, fire characteristics, vegetation, fire weather, modeled smoke patterns, and air pollution concentrations. The prevalence of fire related data and applications on the web is due to a variety of factors including attention given to recent severe wildland fire seasons, the wide range of organizations involved in managing fire and air quality, the multiple uses of the data, a variety of sensors collecting fire data, and the relative simplicity in communicating and understanding the data (e.g., fire location points or satellite images clearly showing the spatial extent smoke plumes).

In many cases, data providers are using standards from the Open Geospatial Consortium (OGC) to improve the ability for others to access and use their data. Three OGC specifications have been used in serving sensor observation data: Web Feature Service (WFS), Web Coverage Service (WCS), and Sensor Observation Service (SOS).

Data Inputs include *in situ*, satellite and aerial data. A data source being investigated to capture the essence of the relationship between sensor webs and forecast models is the use of Unattended Aerial Systems (UAS) for wildfire monitoring. UAS applications illustrate multiple types and locations of data processing along a value added chain that ultimately leads to products used in decision making. UAS driven data chain include multiple levels of data processing. On-board processing on the UAS reduces the amount of information transmitted to the ground. Once on the ground, data are made available through standard interfaces, such as WMS, WCS and as KML files. The served UAS derived data then undergo subsequent processing during data analysis with other data, as input to models, or in validating model output.

Smoke forecasting consists of multiple model components combined to generate predictions of the spatial and temporal extent of smoke and concentrations of air pollutants associated with the smoke. In the Bluesky smoke forecasting framework, independent models for processes such as fuel loading (what is burning), fuel consumed (how much burned), time profile (what is the burn behavior over time), emissions (how much smoke is generated) and dispersion (where does the smoke go) are linked together to create the overall smoke forecast model [7], [8]. The Bluesky framework is in the middle of a redesign to a more modular, flexible and loosely coupled framework suitable for web service applications.

Figure 1 depicts the integration of sensor data with web processing services. In the example, sensors include a UAS from NASA Ames equipped with a wildfire sensor and multiple satellites from which fire locations are derived. The multiple fire location datasets are collected using different algorithms at different times and therefore require some type of reconciliation in order to identify the best estimate of fire locations for use as inputs to a smoke forecast model. A web processing service could encapsulate the reconciliation process and then feed into a series of processing services to execute the smoke forecast.

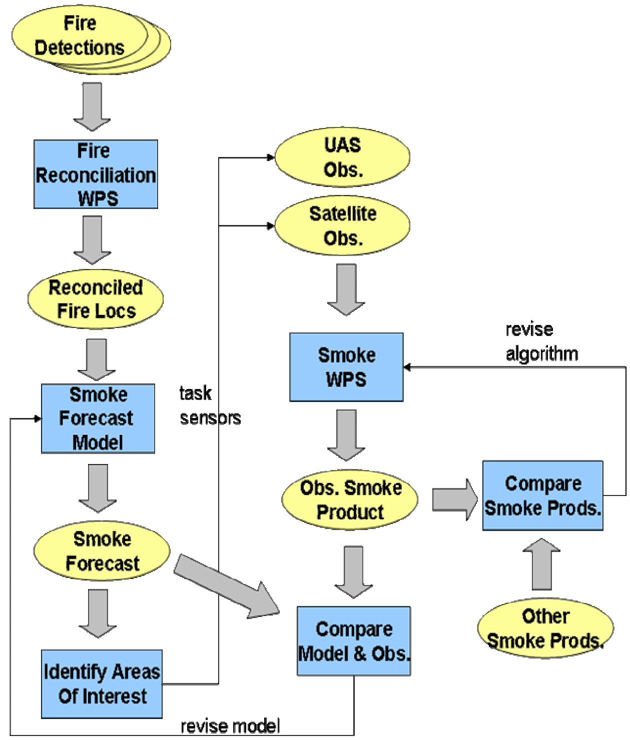

Figure 1. Example flow of multiple sensors and processing services for wildland fire smoke assessment

The smoke forecast identifies areas impacted by smoke 1-3 days in the future and could be used to identify particular times and areas of interest (e.g., urban areas) at which to direct controllable sensors. The output from the newly tasked sensors, such as aboard the NASA EO-1 satellite, could undergo processing to derive a smoke product. The smoke product could be used to validate the smoke forecast model or compared with other smoke products, both of which could be executed as web services. The result is an multi-directional data and processing flow among sensor observations and models aimed at improving each component within the flow as well as generating the best information needed in research and decision making.

5. CONCLUDING REMARKS

This paper presented a general overview of web processing services, including a comparison of two approaches to implementing processing services through web service interfaces. Even though distributed data processing through service oriented frameworks is in the early stages and the standards used to implement such services are still evolving, we submit that it is appropriate to cast forecast model components as web services. This is supported by the fact that many input and validation datasets are already being provided through standard web service interfaces and the processing service and service work flow standards have reached a stage where testing with real forecast models will improve those standards and make them more relevant to the earth science community. Smoke forecast models were highlighted as an initial approach that is being pursued for combining sensor observations and forecast model components in a web service framework.

6. ACKNOWLEDGMENTS

This work was sponsored in part by a NASA ESTO AIST grant and participation in the OGC OWS-5 testbed.

7. REFERENCES

[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/wsdl20-primer/

[2] Schut, P. (ed.) (2007) *OGC® Web Processing Service (WPS)*, OGC 05-007r7. Open Geospatial Consortium Inc.

[3] Falke, S. (ed.) *OGC® OWS-5 Earth Observation Web Processing Services (WPS) Engineering Report*, OGC 08-058r1Open Geospatial Consortium Inc.

[4] Collins, N., G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva, (2005). Design and Implementation of Components in the Earth System Modeling Framework. International Journal of High Performance Computing Applications. Fall/Winter 2005.

[5] Graves, S., Ramachandran, R., Keiser, K., Maskey, M., and C. Lynnes, (2007) Deployable Suite of Data Mining Web Services for Online Science Data Repositories, 87th AMS Conference, San Antonio, TX.

[6] Díaz1, L., Costa, S., Carlos Granell, and M. Gould. (2007) Migrating geoprocessing routines to web services for water resource management applications, 10th AGILE International Conference on Geographic Information Science 2007 Aalborg University, Denmark

[7] Evers, L., S. Ferguson, S. O'Neill, and J. Hoadley. (2006). "Applying BlueSky Smoke Modeling Framework on Wildland Fires." Fire Management Today, Summer 2006, 66 (3), 5-7.

[8] Goodrick, S., Achtemeier G. and L. Yongqiang (2006) Southern Smoke Simulation System - a framework for modeling smoke impacts from prescribed burning in the South , 15th International Emission Inventory Conference, New Orleans, May 15 - 18, 2006.