

WEB BASED GEOPROCESSING USING OPEN SOURCE TOOLS

Kumari Pritee¹, Dr. R.D.Garg², Kapil Oberai³

¹*Research Scholar, Centre for Transportation Systems (CTRANS), IIT Roorkee (India)*

²*Associate Professor, Department of Civil Engineering (Geomatics), IIT Roorkee (India)*

³*Scientist-SD, IIRS, Dehradun, Uttarakhand (India)*

ABSTRACT

As web services has been developed, GIS i.e. Geographical Information System plays a very important role in web based Geoprocessing which is useful for many geospatial projects. An efficient and time saving web based geospatial application compared to earlier application has always been a very big challenge for developers. In this article, aim is to develop such web based Geoprocessing technology which facilitates clients to extract or retrieve any needed information from the map and enables the capability to perform functions like addition, overlay, intersection, spatial analysis etc. in very easier and effective manner from anywhere over the network. In the present implementation, web based Geoprocessing technology has been developed to perform the supporting capability of all GRASS-GIS like overlay, addition and non GRASS-GIS functions in PyWPS which can be Access by clients from anywhere on the network. It is basically developed on Python language, so it is known as a PyWPS (Python web processing service), the service which provides a time saving and fast application compared to PHP WPS application and native support of GRASS-GIS functions. PyWPS has been followed by many steps to develop like request acceptance from web application, run scripts via OpenLayers and provides the user efficient and fast geospatial calculations with no server dependencies. Here the focus is on developing a web based application performing overlay analysis function which makes features of all data set into one dataset and any user can access these computations anywhere over the network.

Keywords :*Apache Server, Grass-GIS 7, Open Layers, QGIS, UBUNTU 10.04 and WPS-Grass-Bridge.*

I. INTRODUCTION

Rapidly increase in use of mapping application over the web developing popular services based on Google maps, facilities and accessibility of data over the network have been increased. All Scientists or researchers are showing their interest in WEBGIS development. A variety of geospatial data and information can be gathered from anywhere. Online Geoprocessing is one of the parts of WEBGIS by which we are not only gathering data but also to extract the information by applying all Grass-GIS functions from anywhere over the network. Geoprocessing means processing of earth information. Geoprocessing is used to create map layer by applying all Grass-GIS and non-Grass-GIS functions. Previously Online Geoprocessing was available on workstations with high performance used for quantitative and qualitative functionalities. But nowadays it is available for

desktop application. Online Geoprocessing offers a variety of calculations, computations and transformations e.g. thematic and temporal information in computation, coordinate transformation etc and all geospatial calculation from large scale to simple calculation e.g. r.add, r.buffer, r.mapcalc etc. This document provides Overlay analysis on PyWPS (Python web processing service) server which is geographic patterns. Open source map layers in a GIS are used to discover relationship among the layers. Overlay analysis is used to search geographic patterns and to determine locations that meet specific criteria. Here we are using Open source tools for online Geoprocessing such as Grass-GIS (The Geographic Resources Analysis Support System), QGIS (Quantum GIS), and WPS (Web Processing Service), Open Layers. In open source software, source code is made available over the web and needs to modify according to new design. Geoprocessing provides Geo-processes in standard way in order to standardize input and output or request and response. It is also called deployment of Geo-processes by implementing some procedure on all Geospatial services like add, multiply, intersect, polygon overlay. Open source is free licensed and any programmers may modify their source code and share within the network. It is standards of OGC (open geospatial consortium) that is non-profit standards organization that is standards for geospatial and location based services. WPS is one of the OGC standards. WPS stands for Web Processing Service. It also makes easy for a client to request the execution of a process and handle output. It provides facilities to encode and decode all process and newly designed process. This implementation focuses on the current OPENGIS specification. The main advantages of using web services that are based on an industry standard framework are a wide variety of implementation support on various platforms, adaptively to document based processing and request-based processing, relative ease to consume other web services, unlimited Geoprocessing functionality supportable and efficient to perform processing the data remotely. WPS defines an interface that facilitates the publishing of geospatial processes and clients discovery of and binding to those processes. This article focused on development of PyWPS (Python Web processing service) server and implementation of it to provide overlay operation over the network without Grass-GIS and build real life based WEBGIS application, which is able to perform for example interpolation of raster data or various digital elevation model analyses. PyWPS is relatively new project (its development started in April 2006). Here PyWPS 3.2.1 has been used. PyWPS as interlayer, which translates requests from the WEBGIS application, runs web server scripts, provides calculation progress and returns the calculation results back to the web browser. PyWPS server provides a framework that makes online Geoprocessing services and calculations over the network faster and easier compared to existing previous methods.

II. LITERATURE REVIEW

Many earlier implementations have been done to acquire better and better performance in the field of online Geoprocessing. This paper introduces one of its implementation in the field of online Geoprocessing named PyWPS. PyWPS was aimed for accessibility of all Grass-GIS modules by all users but this paper introduces PyWPS which would be able to work with the modules of Grass-GIS and non Grass-GIS over the network. PyWPS results in the less time in geospatial calculations than other WPS. Quantitative based availability check and testing for the long running process enables the effective use of web services which is termed as reality check of web servers [1]. A web based multi-tiered distributed geospatial information system has been developed on the platform of GeoBrain web service which is termed as Grass Web application Software system

(GWASS) for reducing the cost of access in the client side [2]. A total object oriented mechanisms, principles and work flow for developing real time web based Geoprocessing service has been covered and explained on the term of automation [3]. The demand of light weight protocols and real time crowd-sourcing capability based Geoprocessing over the web is increasing day by day. These increasing developments have been provided by architecture [4]. Interactive GUI application based on Quality of service aware of Geoprocessing service over the web has been delivered in the form of GeoQOS for standardization and monitoring attributes [5]. Transfer of internet information consisting of temperature of Malaysia into GIS platform has been performed using four modules consisting Google connection, HTML parameters, temperature extraction and Shapefile update module [6].

A Technological Implementation of the new Open Geospatial Consortium Web Processing Service “WPS PHP Server” which provides Web based generalization services and techniques on the real time application that provides a overall accessibility, the Geographic Resources Analysis Support System (GRASS) to generalize roads on the fly [7]. Geospatial calculation and computation on spatial referenced data for overall accessibility has been offered [8] e.g. subtracting one set of spatially referenced numbers from another, or as a global climate change model over the remote servers. US-Geological Survey’s provide a user-centered design of many portable web-mapping clients for online Geoprocessing [9]. Geospatial data has been accessed explicitly for data access standards from the OGC [10]. An investigation about the availability of standard Geoprocessing services and its use in the geosciences domain has been done to find the servers available conforming to the Web Processing [11]. The research goals were (i) Web processing services has been checked whether it is accessible or not, (ii) For long running Geoprocessing processes and web accessible response, this research goal provide quantitative data for extracting different features and (iii) to test the capability for finding Web services in the Domain [12]. The ZOO Project, which is a new open source implementation of the Open Geospatial Consortium’s (OGC) Web Processing Service (WPS), released under the term of the MIT/X-11 license based on a robust server-side C language Kernel (named ZOO Kernel), ZOO Project has been proposed a new way to develop, handle and standardized GIS-based Web services focusing on its assets and limitations, foremost to highlight the new Opportunities provided by such a platform [14]. Plymouth Marine Laboratory (PML) has been developed python WPS implementation (PyWPS) and it has actively supported new developments (SOAP/WSDL) contained in a separated project branch for improvement of WPS-WSDL interface after converting conversion of WPS processes into WSDL integrated easily with Taverna-workbench [19].

III. IMPLEMENTATION

3.1 Presentation

Several implementations were done on the Web Processing Services. Our challenges are to make web processing services more efficient and easy to implement compare with previous implementations. Grass-GIS supported nature of PyWPS (Python enabled web processing service) provides some significant challenges for geospatial calculations. Our initial implementations of PyWPS used overlay function to represent relationships across map layers in GIS. PyWPS may be run in Windows and Linux, but the Windows implementation of PyWPS lacks asynchronous capabilities in get Capabilities due to problems forking all Python processes. Here it

has been efficiently implemented on Ubuntu 10.04. Prerequisites used are Python 2.5, 2.6 and python-xml. A couple of dependencies such as libapache2-mod-python, python-htmtempl, and python-magic have been installed. Packages has been used which is essential for the development. Packages used are Apache Web Server, GRASS (Geographic Resources Analysis Support System), PROJ.4 (Cartographic Projection Library), and GDAL/OGR (Geospatial Data Abstraction Library). Latest PyWPS has been downloaded [PyWPS-3.2.1]. Connection between apache server and PyWPS has been established. It provides a server-side C kernel that makes creation and management of web services by loading dynamic libraries possible. It also connects to geospatial libraries from which geospatial calculation possible. Here web services have been programmed in python Web-scripting language but it could be possible with C, Python, Java, FORTRAN, PHP and JavaScript. It updates existing code to create new web services. PyWPS distribution does not include any default processes. For any other process, client has to create a custom process and add this process to PyWPS GetCapabilities.

3.2 Architecture

In this section architecture has been explained in Fig 1. Service provider is the key concept of this architecture that has been explained in this section.

This architecture run on Apache web server which handles all traffic related to network and write script on python. This Python script includes HTTP request and response to any input POST data.

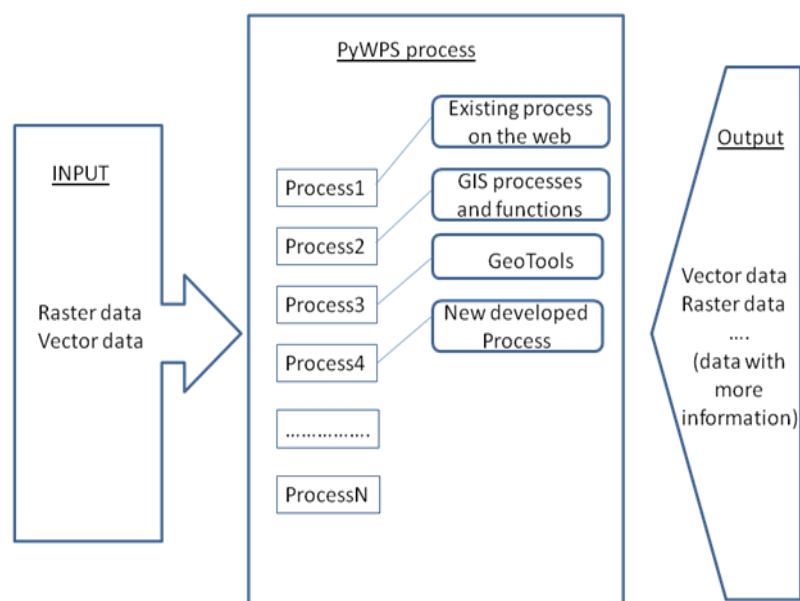


Figure 1 Basic PyWPS process

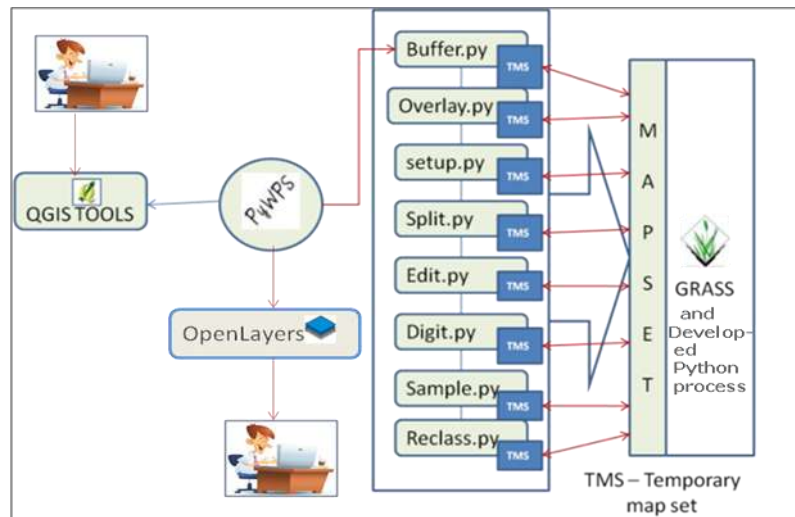


Figure 2 Architecture of Online Geoprocessing using PyWPS

Our implementation is based on architecture given in Fig. 1. In Fig. 2, a complete Grass-GIS process has been described on PyWPS. PyWPS includes following steps: 1) Import custom process created on python from PyWPS into Grass-GIS function e.g. v.in.ogr in vector region. 2) Region should be set raster e.g. r.in.gdal in raster region 3) Export that process back to PyWPS to execute that process and to provide the response e.g. v.out.ogr in vector or r.out.gdal in raster.

After going through this architecture, we need to initialize each incoming request separate, since HTTP requests are stateless. First, XML file should be located and parsed. Here configuration.xml files are located and parsed at the beginning of the initialization. XML (Extensible Markup Language) is used for providing information about the working directory i.e. from where it can find and somewhat Grass-GIS set up also. WPS services are accessible via two HTTP methods (GET and POST). Fig. 3 shows the HTTP methods and their work flow.

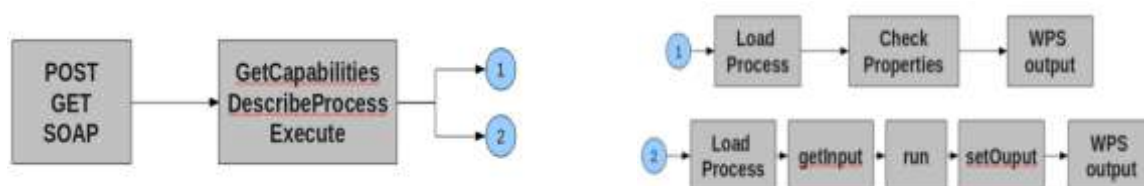


Figure 3 Description of HTTP Methods

To parse the input, handler class is necessary like GET handler, for Get Parser which makes available all the function key and value pairs that provides variables. The Post Parser parses the request written in XML.

3.3 PyWPS as a Framework Provider

A PyWPS service provider is defined as a group of Services Shared Objects and one common gateway interface file that is wrapper file (pywps.cgi). PyWPS is normally run as a CGI script (pywps.cgi) but it can also be run using mod_python and support for mod_wsgi (Web Server Gateway Interface) is being finalized. Checking of

all the possibilities of CGI script integration in python shell play an important role in WEBGIS clients. A WebGIS interface can improve better understanding for area information for both before and after data processing and analysis. Since the code is written in standard Python it can also be ported as a Java let (using Jython) and run from Tomcat application server. PyWPS is lightweight, with only 5000 lines of code, which makes it a good candidate for testing new concepts This CGI file provides all the metadata configuration of all processes, description and their execution. Service provider facilitates the Get Capabilities and Describe Process Requests. Pywps.cgi file defines the configuration files which checks whether all required inputs were provided in the request. Here we have been implemented two kinds of Processes in the PyWPS Server: GRASS and native processes. Grass Processes uses GRASS GIS functions to do the implementation. Native processes do not use GRASS GIS functions for implementation. It builds one module to develop geospatial functions without using GRASS GIS functions. Module is a set of processes. Initialization of PyWPS Server helps in defining each module by XML file. There are two additional steps needed to make a Grass-GIS process run because the Grass-GIS processes are separated. These two extra steps are listed below:-

- The transferred (or referenced) data sets needs to be imported to form a GRASS GIS database.
- The results have to be exported after processing and querying has been done in a proper way. The care should be taken that only Geographic Markup language is supported.

No programming is needed for adding the process. The only requirement is to connect a process written in XML to a Grass-GIS module. Suppose a more complex process is needed, then the requirement is to write a script for chaining multiple Grass-GIS modules. There is no any restriction for programming language used. Execution of process or sequence diagram can be represented by Fig. 4.

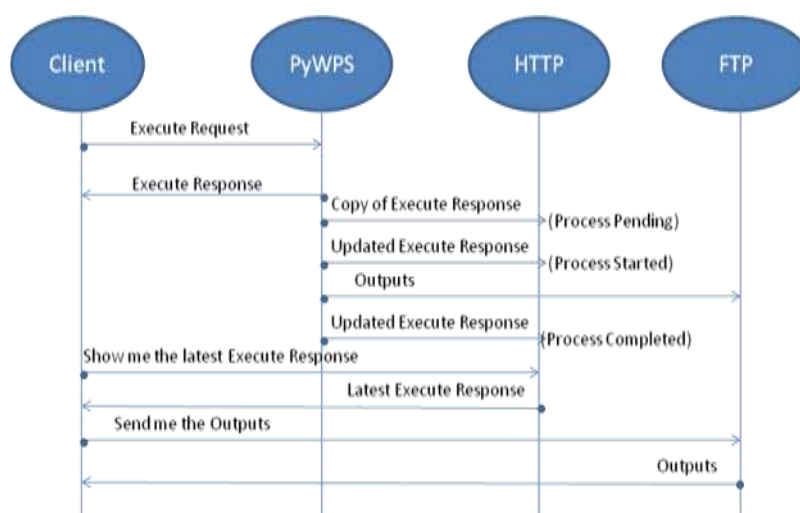


Figure 4 Execution of process through request and response.

3.4 OSGeo libraries for PyWPS server

It is basically used in a form of platform for numerous and effective libraries for connecting together and able to make easy access for all web services. The GDAL/OGR library is used to perform almost all raster and vector functions written in python with supported libraries.

GDAL/OGR

GDAL (Geospatial Data Abstraction Library) is basically supported for various functions in projects e.g. importing, exporting, convert reproject for all data in raster and vector format. This library is used in set up of very efficient web services by using both single and multiple geometries spatial operations. Gdalgrid and gdaltranslate capabilities in WPS implementation is also possible .In case of Map Server, GDAL and OGR are also needed.

GRASS GIS 7

It refers to Geographical Resources Analysis Support system. It is one of the efficient Open Source GIS software that export process description of WPS processes returning Grass-GIS functions with XML scripting. PyWPS is written in native support of GRASS and its functions. All Grass-GIS function with python scripting are termed as PyWPS Service Provider. Several successful examples have been carried out using the r.add, r.mapcalc, r.div, r.mult and r.sub functions.

PROJ.4 (Cartographic Projection Library)

This library is basically used for almost all open source projects for data transformation from one coordinate to another coordinate. Proj4 is especially beneficial for integration of Map Server as well, in python-pyproj package.

QGIS (Quantum Geographic Information System)

It refers as an open source desktop application that runs on different platforms and freely available providing all capabilities of data viewing, editing and analysis functions. Its function is similar to other software like Grass-GIS. It allows users to create maps with different raster or vector layers in different map projections. It supports different kind of raster images and vector images. Vector image is stored as point, line or polygon and raster image needs to be georeferenced by the software. To illustrate a workflow sequence that triggers a PyWPS-based model, a PyWPS instance is created and a set of computational geometries is to be simplified with the help of this instance. All request and response depends upon XML messages generated by a web client application. For this, user should know the entry point of PyWPS instance. Such entry point can be retrieved from the web services but can get more and more information by using queries on service metadata via GetCapabilities request. Retrieving the metadata such as entry point is important for further communication called Operations Metadata. By these services, Individual information can also get about the provider called Service Provider and many models are offered called Process Offerings. Fig. 5 shows the overlay process of helloworld example.

[http://localhost/cgi-bin/wps.py?Service=WPS&request=execute&version=3.2.1&identifier=helloWorld&DataInputs=\[message=Tim\]](http://localhost/cgi-bin/wps.py?Service=WPS&request=execute&version=3.2.1&identifier=helloWorld&DataInputs=[message=Tim])

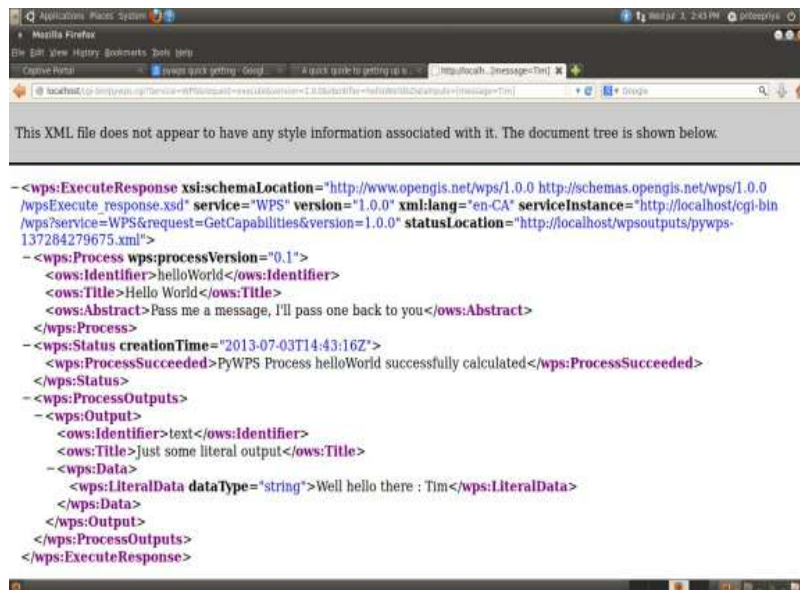


Figure 5 Execution of helloworld.py process

Open Layers

Open Layers (OL) is an open source JavaScript API that represents a map dynamically in any web page displaying map tiles and markers loaded from any source.

The API is extensive and offers the following functionalities:

- Availability and usage of mostly web based GIS application e.g. Google maps, Open street etc;
- Support of map servers like GeoServer or Map Server;
- Easily Support for both Web Map Service and WFS specification with common Image formats (in PNG, GIF or JPEG format);
- Support for all raster as well as vector formats: GML, GeoTIFF, KML, OGC, img etc.

PyWPS supports future developments concerning WPS integration and OL (Open Layers) should use the API's native functionalities. Earlier Plymouth Marine Laboratory has developed some examples based on the PyWPS JavaScript client that provides support for the WPS operations, HTTP request and response and XML processing.

WPS implementation via OL has two strategies:

- Toolbox, where user uses Get Capabilities / describe Process to determine which process to run;
- Pre-defined process that will be run when necessary (e.g. an event that triggers a JavaScript call to WPS).

Finally, a new WPS-Grass-Bridge has been focused that makes easily importing of Grass-GIS modules into PyWPS in a convenient way and allows easy integration into QGIS-WPS Plug-in.

The WPS-Grass-Bridge depends on Grass-GIS 7 trunk, PyWPS trunk and PyXB 1.1.2. So, we have been used the latest versions of PyWPS and GRASS GIS 7.

The general principle is

- Install GRASS GIS trunk version 7 and PyWPS 3.2.1 trunk via svn.

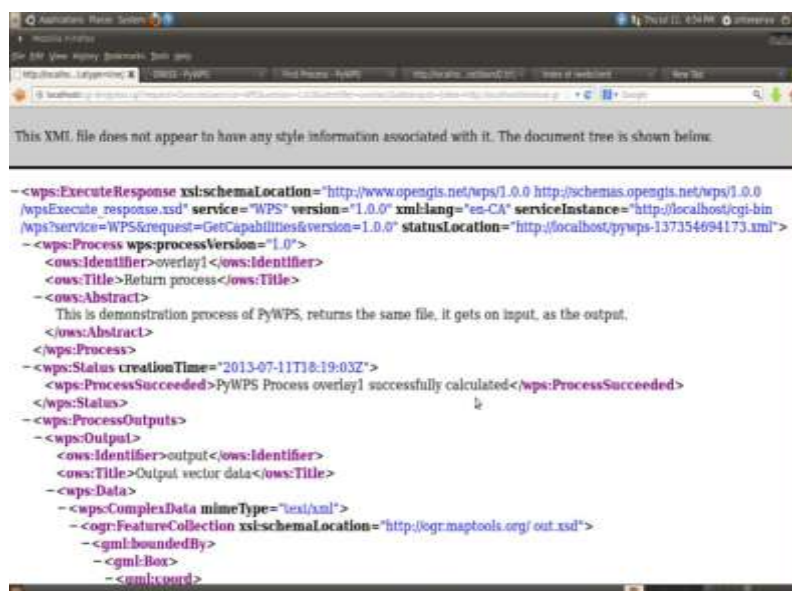
- Install PyXB, which facilitates to generate the PyWPS processes.
- Download the Latest WPS-Grass-Bridge.
- Modify the GlobalGrassSettings.py script and check PyWPSGrassModuleStarter.py.
- Create the XML process description of all processes for each Grass-GIS module which need to be attached.
- Based on the XML files generate the PyWPS processes with GrassXMLtoPyWPS.py.
- Provide the WPS-Grass-Bridge and the generated processes into a PyWPS process directory.

IV. RESULTS

The use of libraries like GDAL/OGR, PROJ4 in a standard way on the basis small changes in original codes and the native support for GRASS GIS makes PyWPS more effective than other web processing services. Fig. 6 shows an overlay process on PyWPS server. Vector operations on a single geometry can also be performed e.g. boundary, centroid. After Get Capabilities and describe Process, Execute operation on a given processes has been performed and the final response will be displayed via XML scripting and graphically displayed on map simultaneously.

[http://localhost/cgi-](http://localhost/cgi-bin/grass.cgi?request=Execute&service=WPS&version=1.0.0&identifier=overlay1&datainputs=[data=http://localhost/landuse.gml;mask=http://localhost/file2.gml;atype=line])

[bin/grass.cgi?request=Execute&service=WPS&version=1.0.0&identifier=overlay1&datainputs=\[data=http://localhost/landuse.gml;mask=http://localhost/file2.gml;atype=line\]](http://localhost/landuse.gml;mask=http://localhost/file2.gml;atype=line)



```

This XML file does not appear to have any style information associated with it. The document tree is shown below.
- <wps:ExecuteResponse xsi:schemaLocation="http://www.openengis.net/wps/1.0.0 http://schemas.openengis.net/wps/1.0.0/wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-CA" serviceInstance="http://localhost/cgi-bin/wps?service=WPS&request=GetCapabilities&version=1.0.0" statusLocation="http://localhost/pywps-137354694173.xml">
- <wps:Process processVersion="1.0">
  <ows:Identifier>overlay1</ows:Identifier>
  <ows:Title>Return process</ows:Title>
  <ows:Abstract>
    This is demonstration process of PyWPS, returns the same file, it gets on input, as the output.
  </ows:Abstract>
</wps:Process>
- <wps:Status creationTime="2013-07-11T18:19:03Z">
  <wps:ProcessSucceeded>PyWPS Process overlay1 successfully calculated</wps:ProcessSucceeded>
</wps:Status>
- <wps:ProcessOutputs>
- <wps:Output>
  <ows:Identifier>output</ows:Identifier>
  <ows:Title>Output vector data</ows:Title>
- <wps>Data>
  - <wps:ComplexData mimeType="text/xml">
    - <ogr:FeatureCollection xsi:schemaLocation="http://ogr.maptools.org/out.xsd">
      - <gml:boundedBy>
        - <gml:Box>
          - <gml:coord>

```

Figure 6 Execution of Overlay.py operation on PyWPS

Now the way such PyWPS Web services called have been explained and make easily access from a client-side Web mapping application based on Open Layer. Fig. 7 shows the web mapping application of overlay operation on the client side.

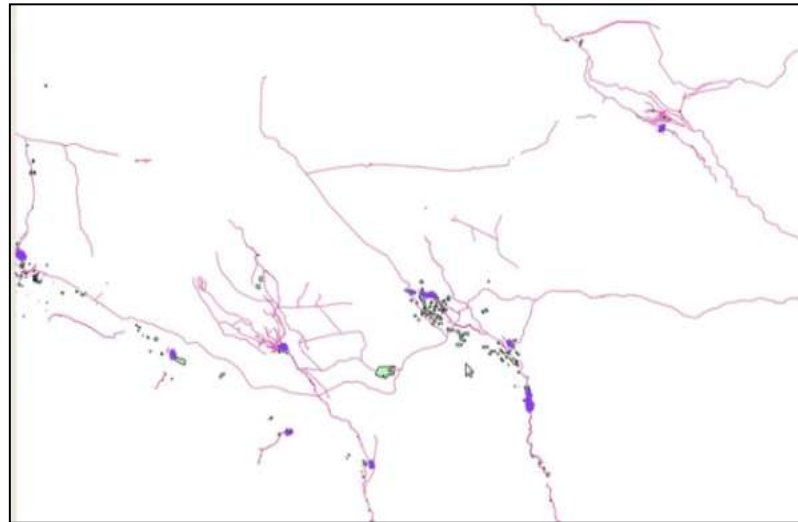


Figure 7 Client-side web mapping application of Overlay Operation

The user can also select another polygon to perform a multi-geometry operation after overlaying of maps is shown in the output. Future developments also depend upon the integration of all open source GIS and non-GIS libraries.

REFERENCES

Journal Papers:

- [1] F. J. Lopez-Pellicer, W. Renteria-Agualimpia, R. Bejar, P. R. Muro-Medrano and F. J. Zarazaga-Soria, Availability of the OGC Geoprocessing standard: March 2011 reality check, *Computers & Geosciences* 47, Elsevier Ltd (copyright), 2012, pp. 13–19.
- [2] F. Qiu, F. Ni, B. Chastain, H. Huang, P. Zhao, W. Han and L. Di, GWASS : GRASS web application software system based on the GeoBrain web service, *Computers & Geosciences* 47, 2012, pp. 143–150.
- [3] S. Shi and N. Walford, Automated Geoprocessing Mechanism, Processes and Workflow for Seamless Online Integration of Geodata Services and Creating Geoprocessing Services, *IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*, VOL. 5, NO. 6, 2012, pp. 1659-1664.
- [4] P. Zhao, T. Forester and P. Yue, The Geoprocessing Web, *Computers & Geosciences* 47, Elsevier Ltd (copyright), 2012, pp. 3–12.
- [5] P. Yue, Z. Tan and M. Zhang, GeoQOS : Delivering Quality of Services on the Geoprocessing Web, *Proceedings Template - WORD - FOSS4G*, 2014, pp. 59-70.
- [6] M. I. Sameen, A. A. Ali, N. O. A. Wahid and M. A. A. Kubaisy, An Approach to Develop a Geographic Information Database Using Dot Spatial Open Source Platform and Google Search Engine, *Computer Science Systems Biology* vol. 7, 2014, pp. 217-220.

Proceedings Papers:

- [7] J. Cepicky, P. Gnip, S. Kafka, I. Koskova, Geospatial data management and integration of geospatial web services, World Conference on Agriculture Information and IT, 2008, PP. 1-16.
- [8] B. Schaeffer, Towards a Transactional Web Processing Service (WPS-T), Proc 6th GI Days, Institute for Geoinformatics, Muenster, 2008, pp 91–117.
- [9] W. Bergenheim, L. T. Sarjakoski, T. Sarjakoski, A Web Processing Service for GRASS GIS to provide on-line Generalization, 12th AGILE International Conference on Geographic Information Science, Germany, 2009, pp. 1-10.
- [10] G. Percivalla, L. Menardb, L. K. Chungc, S. Nativi and J. Pearlmane, Geoprocessing in cyber infrastructure making the web an easy to use Geospatial computational platform, 34th International Symposium on Remote Sensing of Environment, Australia, 2011, pp. USBkey.
- [11] F. J. Lopez-Pellicer, W. Renteria-Agualimpia, R. Bejar, J. Valino, F. J. Zarazaga-Soria, P. R. Muro-Medrano, Implantation of OGC Geoprocessing services for Geoscience, Proceedings of the II Iberian Conference of Spatial Data Infrastructures, 2011, pp. 1-12.
- [12] E. B. Wolf and K. Howe, Web client Based Distributed Generalization and Geoprocessing, International Conference on Advanced Geographic Information Systems & Web Services, 2012, pp. 123-128.

Web References:

- [13] <http://www.zooproject.org>. Gerald Fenoy, ZOO-Project: the open WPS platform, Applied Geomatics, 2012.
- [14] <http://wpsint.tigris.org>. OGC Web Processing Service (WPS) interface, 2012.
- [15] http://2010.foss4g.org/presentations_gensel.php. Open Geospatial Consortium Inc. OpenGIS® (FOSS4G 2010) Presentations about WPS at Source Geospatial Foundation (OSGeo), 2010.
- [16] <http://pywps.wald.intevation.org>. Conference on Advanced Geographic Information Systems and Web Services, PyWPS, 2012.
- [17] http://wiki.ieee-earth.org/Documents/GEOSS_Tutorials. Steps for Plymouth Marine Laboratory implementation of Python WPS.
- [18] www.osgeo.org.