GRASS goes web: PyWPS 0.1

J. Čepický

1Mendel University of Agriculture and Forestry Brno
   Faculty of Forestry and Wood Technology
   Department of Geoinformation Technologies

FOSS4G2006, Lausanne, Switzerland
TOC

1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion
Open Geospatial Consortium, Inc (OGC)

- International voluntary consensus standards organization
- Development and implementation of standards for geospatial content and services
- http://www.opengeospatial.org/
Open Geospatial Consortium
OpenGIS® Web Processing Service
PyWPS 0.1
PyWPS in action
Conclusion

Web service

TOC

1 Open Geospatial Consortium
   • Web service

2 OpenGIS® Web Processing Service
   • GetCapabilities
   • DescribeProcess
   • Execute

3 PyWPS 0.1
   • How does it work
   • Sample Process

4 PyWPS in action
   • WPS Demo
   • Precision farming

5 Conclusion

ˇCepický

Čepický

GRASS goes web: PyWPS 0.1
Web service

- Software system designed to support interoperable machine-to-machine interaction over a network (Wikipedia)
- In OGC terminology, „Service“ refers to a processing task that is invoked by a client and executed by a server, usually across a network.
- The OpenGIS Specifications that make this possible are referred to as "OGC Web Services."
- OpenGIS Web Service (OWS):
  - OpenGIS Catalog Service (CAT)
  - OpenGIS Web Coverage Service (WCS)
  - OpenGIS Web Feature Service (WFS)
  - OpenGIS Web Map Service (WMS)
  - …
  - Web Processing Service (WPS) (draft)
TOC

1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion

Čepický
GRASS goes web: PyWPS 0.1
OpenGIS® Web Processing Service

- Document OGC 05-007r4, version 0.4.0
- Not yet OGC standard, „Discussion Paper“, Draft
- To offer any sort of GIS functionality to clients across a network.
- XML-based communication protocol.
OpenGIS® Web Processing Service

- Document OGC 05-007r4, version 0.4.0
- Not yet OGC standard, „Discussion Paper“, Draft
- To offer any sort of GIS functionality to clients across a network.
- XML-based communication protocol.

http://www.bnhelp.cz/cgi-bin/wps?service=WPS&version=0.4.0&request=...
1 Open Geospatial Consortium
   - Web service

2 OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3 PyWPS 0.1
   - How does it work
   - Sample Process

4 PyWPS in action
   - WPS Demo
   - Precision farming

5 Conclusion

TOC
request=GetCapabilities

http://www.bnhelp.cz/cgi-bin/wps?service=WPS&version=0.4.0&
request=GetCapabilities

<?xml version="1.0" ?>
<Capabilities version="0.4.0" ...
<ows:ServiceIdentification>
   <ows:Title>Sample WPS server</ows:Title>
   <ows:Abstract>WPS for Lausanne</ows:Abstract>
   <ows:ServiceType>WPS</ows:ServiceType>
   <ows:Fees>free</ows:Fees>
</ows:ServiceIdentification>
<ows:ServiceProvider>
   <ows:ProviderName>GDF</ows:ProviderName>
   <ows:ServiceContact>
      <ows:IndividualName>Jachym Cepicky</ows:IndividualName>
      <ows:PositionName>Student</ows:PositionName>
   ... 
</ows:ServiceContact>
</ows:ServiceProvider>
request=GetCapabilities

```xml
<ProcessOfferings>
  <Process processVersion="0.1">
    <ows:Identifier>addvalue</ows:Identifier>
    <ows:Title>Add some value to raster map</ows:Title>
  </Process>
  <Process processVersion="0.1">
    <ows:Identifier>classify</ows:Identifier>
    <ows:Title>Image classification</ows:Title>
    <ows:Abstract>
      GRASS processed imagery classification. Only unsupervised is supported at the moment.
    </ows:Abstract>
  </Process>
  <Process processVersion="0.1">
    <ows:Identifier>shortestpath</ows:Identifier>
    <ows:Title>Shortest path</ows:Title>
  </Process>
</ProcessOfferings>
```
request=DescribeProcess

http://www.bnhelp.cz/cgi-bin/wps.py?service=WPS&
version=0.4.0&
request=DescribeProcess&identifier=addvalue

```xml
<?xml version="1.0" ?>
<ProcessDescriptions ...>
   <ProcessDescription ...>
      <ows:Identifier>addvalue</ows:Identifier>
      <ows:Title>Add value</ows:Title>
      <ows:Abstract>Adds some value to each cell of input raster map</ows:Abstract>
      <DataInputs>
         <Input>
            <ows:Identifier>value</ows:Identifier>
            <ows:Title>Value to be added</ows:Title>
            <LiteralData>
               <AllowedValues>
                  <Value>1</Value>
                  ...
               </AllowedValues>
               <ows:DefaultValue>10</ows:DefaultValue>
            </LiteralData>
         </Input>
         <Input>
            <ows:Identifier>map</ows:Identifier>
            <ows:Title>Input raster map</ows:Title>
            <ComplexData defaultFormat="image/tiff">
               ...
            </ComplexData>
         </Input>
      </DataInputs>
   </ProcessDescription>
</ProcessDescriptions>
```
request=DescribeProcess

```xml
<ProcessOutputs>
  <Output>
    <ows:Identifier>value</ows:Identifier>
    <ows:Title>literal value + 1</ows:Title>
    <LiteralOutput>
      ...
    </LiteralOutput>
  </Output>
  <Output>
    <ows:Identifier>map</ows:Identifier>
    <ows:Title>Resulting output map</ows:Title>
    <ComplexOutput defaultFormat="image/tiff">
      ...
    </ComplexOutput>
  </Output>
</ProcessOutputs>
</ProcessDescription>
</ProcessDescriptions>
```
request=Execute

http://www.bnhelp.cz/cgi-bin/wps.py?service=WPS&
version=0.4.0&
request=Execute&identifier=addvalue
DataInputs=value,5,map,http://localhost/data/soils.tif

<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<Execute service='wps' version='0.4.0' store='true' status='false'
 xmlns="http://www.opengeospatial.net/wps"
 xmlns:ows="http://www.opengeospatial.net/ows">
<ows:Identifier>addvalue</ows:Identifier>
<DataInputs>
 <Input>
  <ows:Identifier>value</ows:Identifier>
  <LiteralValue>5</LiteralValue>
 </Input>
 <Input>
  <ows:Identifier>map</ows:Identifier>
  <ComplexValueReference
   reference="http://localhost/data/soils.tif"/>
 </Input>
 ... 
</DataInputs>
</Execute>
Responce XML

```xml
<?xml version="1.0" ?>
<ExecuteResponse ...
  <ows:Identifier>addvalue</ows:Identifier>
  <Status>
    <ProcessSucceeded/>
  </Status>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>value</ows:Identifier>
      <ows:Title>literal value + 1</ows:Title>
      <LiteralValue>6</LiteralValue>
    </Output>
    <Output>
      <ows:Identifier>value</ows:Identifier>
      <ows:Title>Resulting output map</ows:Title>
      <ComplexValueReference
        format="image/tiff"
        ows:reference="http://www.bnhelp.cz/wpsoutputs/output2-2006-8-21-14-54-42.tif"
      />
    </Output>
  </ProcessOutputs>
</ExecuteResponse>
```
Open Geospatial Consortium
OpenGIS® Web Processing Service
PyWPS 0.1
PyWPS in action
Conclusion

How does it work
Sample Process

1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion

ˇČepický
GRASS goes web: PyWPS 0.1
PyWPS 0.1

- Implementation OGS’s WPS Standard (90-95 %)
- CGI Application, KISSed
- Python programming language
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)

- CLI
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)

- CLI
- More than 300 modules for raster and vector analysis
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)
  - CLI
  - More than 300 modules for raster and vector analysis
  - GNU/GPL
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)

- CLI
- More than 300 modules for raster and vector analysis
- GNU/GPL
- GRASS Functionality can be via PyWPS offered in Internet
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)

- CLI
- More than 300 modules for raster and vector analysis
- GNU/GPL
- GRASS Functionality can be via PyWPS offered in Internet

- User does not need Desktop-GIS (GRASS, ESRI, Idrisi, ...) – Web browser becomes GIS
PyWPS 0.1

The „engine“ of PyWPS is GIS GRASS (>= 6.1)

- CLI
- More than 300 modules for raster and vector analysis
- GNU/GPL
- GRASS Functionality can be via PyWPS offered in Internet

- User does not need Desktop-GIS (GRASS, ESRI, Idrisi, . . .) – Web browser becomes GIS

- One can use other CLI-oriented programs (PROJ.4, GDAL, R, . . .)
PyWPS 0.1 – Execute

1. Controlling input data, if all necessary parameters have arrived (Identifier, DataInputs, ...)
2. Loading process, for each input:
   - LiteralValue: Controlling, if input fits AllowedValues array
   - ComplexValue: Embed input files will be extruded from input XML request into separate files.
   - ComplexValueReference: Tries to download the data from external source and stores it to new file
   - BoundingBoxValue
3. If some DataInput is missing, it looks for the default value value
PyWPS 0.1 – Execute

5. Creates temporary GRASS Location or just temporary Mapset within existing location, which will be deleted, after the work is done

6. Calls function `execute()` of the process

7. Formulates output XML file

8. Deletes temporary files (location, mapset, pid file)

9. Returns output XML or resulting map file (TIFF, GML) to the client

10. Process can be run asynchronously: After the request is accepted, XML response is immediately returned with `<ProcessAccepted />` element and the calculation is forked to background.
Addvalue - Sample process

- **Inputs**
  - Literal input: Value to be added
  - ComplexValueReference input – Some raster map

- **Outputs**
  - Literal output – Input+1
  - ComplexValueReference: Resulting raster map (GeoTIFF)
Sample Process

```python
01 class Process:
02     def __init__(self):
03         self.Identifier = "addvalue"
04         self.Title="Sample process for demonstration purposes"
05         self.Inputs = [
06             {'Identifier': 'value',
07             'Title': 'Value to added',
08             'LiteralValue': {'values':[0,1,2,3,4,5]},
09             'dataType': type(0),
10             'value': 0, # default},
11             {'Identifier': 'map',
12             'Title': 'The raster map',
13             'ComplexValueReference': {'Formats': ['image/tiff']},
14             'value': "output.tif" # default can be set
15         ],
16         self.Outputs = [
17             {'Identifier': 'value',
18             'Title': 'Input value + 1',
19             'LiteralValue': {}},
20         ],
21         self.storeSupported = "true" # output data can be stored on the server
22         self.statusSupported = "true" # process can run asynchronously
```
Sample Process

```python
29   def execute(self):
30       self.status = ['The start', 5]
31       self.Outputs[0][value] = self.Inputs[0][value]+1
32       self.status = ['LiteralValue set', 20]
33
34       self.status = ['Data import', 25]
35       os.system('r.in.gdal in=%s out=map' % (self.Inputs[1][value]))
36
37       self.status = ['Creating output map', 50]
38       os.system('r.mapcalc map=map+%d' % (self.Inputs[0][value]))
39
40       self.status = ['Exporting map', 75]
41       if os.system('r.out.gdal in=map out=output.tif type=UInt16 >
42           return "Could not export map"
43       else: # ok
44           return
```
1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion
Open Geospatial Consortium
OpenGIS® Web Processing Service
PyWPS 0.1
PyWPS in action
Conclusion

WPS Demo
Precision farming

TOC

1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion
WPS Demo

- Web Client provided by Help Service Remote Sensing
- GIS GRASS performs all geographical operations
- PyWPS as glue between them
WPS Demo

Shortest path ($v$.net$.path$)
Unsupervised image classification (i.maxlik)
Unsupervised image classification (*i.maxlik*)
Unsupervised image classification (**i.maxlik**)

Čepický  
GRASS goes web: PyWPS 0.1
WPS Demo

Line-of-sight ($r_{los}$)
WPS Demo

Flow analysis (Jizerské Mountains) (r.flow)
1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion
Precision farming

Used operations:

- Data import from PostgreSQL
- Interpolation of nutrients decomposition
- Calculation of fertilization recommendation
- ...

- v.in.ogr, r.srf.rst, r.mapcalc, r.reclass, r.stats, SQL, ...
Precision farming

Fields and pH decomposition
Precision farming

Culture
Precision farming

Fertilization variation
Precision farming

Fertilization variation - just for selected fields
1. Open Geospatial Consortium
   - Web service

2. OpenGIS® Web Processing Service
   - GetCapabilities
   - DescribeProcess
   - Execute

3. PyWPS 0.1
   - How does it work
   - Sample Process

4. PyWPS in action
   - WPS Demo
   - Precision farming

5. Conclusion
Conclusion

+ WPS Standard implemented to usable degree
+ Making GRASS scripts run via web-interface was never easier
+ It is relatively simple to connect UMN MapServer (or ARC IMS) with GRASS via PyWPS. Further GRASS development will make this even easier.
Further development

- Process definition (data inputs and outputs) is primitive – build set of classes for process definition
- Implementation of new GRASS-python interface (Alessandro Frigeri aka ’geoalf’)
- 3D views via VTK (Sören Gebbert aka ’huhabla’)
- …
- Bug fixing
End

jachym.cepicky@centrum.cz
http://Les-ejk.cz

svn checkout https://subversion.gdf-hannover.de:8080/svn/pywps/trunk

Development of PyWPS was financially supported by Deutsche Bundesstiftung Umwelt (http://dbu.de), GDF Hannover and Help Service Remote Sensing were contributed. Presentation was made possible thanks to GA ČR project nr. 526/03/H036 „Current stage and trends of development of forests in cultural landscape“.