The WPS 2.0 Standard

Benjamin Pross, Matthias Müller
FOSS4G-E, 15.07.2014
Overview

• Status update
• WPS 2.0 components
• Conformance tests
WPS 2.0 Status

- SWG vote two weeks ago
- OAB telecon today 9:30 p.m.
- Public comment period will start soon (hopefully)
- Still draft status
WPS 1.0

• Has gained some attention
  - Recently also in the environmental modeling domain
• Long due overhaul (dates back to 2007)
• Lot's of change requests accumulated
• Lacks support for process cataloguing
• No Abstract Test Suite (ATS)
WPS 2.0 - Components

• Conceptual Model - Core
• (Native) Process model / data model
• Support for foreign process models
• Common Service operations
• Process profiles
WPS 2.0 - The core

- Is a conceptual model
- Formulates minimal platform-independent requirements for a WPS
- Can be realized in different Distributed Computing Platforms / Architectures (SOA, REST, ...)

WPS 2.0 - The core

- **Web Processing Service** provides process
- **Job Control Operation** provides control & monitor
- **Process** runs job
- **Job** has data & status
- **Data**
- **Status**
WPS 2.0 - The core

- Requirements around
  - Service discovery
  - Service capabilities
  - Abstract process model
  - Job control
  - Process execution
  - Data transmission by value / by reference
  - Job monitoring
WPS 2.0 - Process Model

- Process interface / signature
- Descriptive elements
- Data types
Process Model WPS 1.0

Y = f(X)

Input 1: Dataset A
Input 2: Dataset B
Input n: Dataset C
Dataset D
Dataset E
Dataset F

Output 1: Dataset G
Output n: Dataset H
Process Model WPS 2.0

- Process (@Identifier, @Title, @Abstract, @keywords)
  - Input[0..*]
  - Input[0..*]
    - Input[0..*]
  - ... 
  - ... 
  - Output[1]
    - Output[1..*]
    - ... 
  - Output[1]
WPS 2.0

- Nested in-/outputs
  - Specify one or more input types as „data type“ of an input
  - Established relations between inputs
  - Allows more structured inputs
  - Nesting level should be kept low

```xml
<DataInputs>
  <Input minOccurs="1" maxOccurs="50">
    <ows:Identifier>VALUE_WEIGHT_PAIRS</ows:Identifier>
    <ows:Title>...</ows:Title>
    <ows:Abstract>...</ows:Abstract>
  </Input>
  <Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>VALUE</ows:Identifier>
    <ows:Title>...</ows:Title>
    <ows:Abstract>...</ows:Abstract>
  </Input>
  ...
</DataInputs>
```
Process Model WPS 2.0

**DescriptionType**
- `title[1]`: ows:Title
- `abstract[0..1]`: ows:Abstract
- `keywords[0..1]`: ows:Keywords
- `metadata[0..*]`: ows:Metadata

**Process**
- `Input[0..*]`: Input
- `Output[1..*]`: Output

**Input**
- `minOccurs[1]`: nonNegativeInteger
- `maxOccurs[1]`: positiveInteger

**Output**
- `Output[0..*]`
Process Model WPS 2.0

- More symmetry for inputs and outputs
- Documentation links in metadata elements

```
<ows:Metadata
    xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
    xlink:href="http://my.wps.server/processes/proximity/Planar-Buffer.html#input_geometry"/>

<ows:Metadata
    xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
    xlink:href="http://some.host/profileregistry/implementation/Planar-GML-Buffer.html"/>
```
Data model for Process I/O

- IOData
  - ComplexData
  - LiteralData
  - BoundingBoxData

- generic
- pre-defined
Foreign process models

- Idea triggered by SensorML change request
- Purpose: WPS protocol shall support other process models that have their own descriptive model and data types
- Solution: Abstract minimum requirements for process models that shall be used in conjunction with WPS
  - Identifiers for processes, inputs, outputs
  - Well-defined data types
  - Cardinality constraints
  - ...

The WPS 2.0 Standard (preliminary information), 2014-07-15, FOSS4G-E, Bremen
Common service operations

- GetCapabilities
- DescribeProcess
- Execute (mode=sync|async|auto)
- GetStatus (async)
- GetResult (async)
- Dismiss (extension)
  - Cancel / release running job
  - Release server-stored results
GetCapabilities

• More operations

```xml
<wps:Contents>
  <wps:ProcessSummary
    jobControlOptions="sync-execute async-execute dismiss">
    <ows:Title>Euclidean Distance</ows:Title>
    <ows:Identifier>
      http://my.site/distance-transform/euclidean-distance
    </ows:Identifier>
  </wps:ProcessSummary>
  <wps:ProcessSummary
    jobControlOptions="async-execute dismiss">
    processVersion="1.4.0">
    <ows:Title>Cost Distance</ows:Title>
    <ows:Identifier>
      http://my.site/distance-transform/cost-distance
    </ows:Identifier>
  </wps:ProcessSummary>
</wps:Contents>
```
DescribeProcess

• More metadata

<wps:ProcessOffering>
  jobControlOptions="sync-execute async-execute dismiss"
  outputTransmission="value reference">

  <wps:Input>
    <ows:Title>Geometry to be buffered</ows:Title>
    <ows:Abstract>
      Simple Features geometry input in GML or GeoJson
    </ows:Abstract>
    <ows:Identifier>INPUT_GEOMETRY</ows:Identifier>
    <wps:ComplexData>
      <wps:Format mimeType="text/xml" encoding="UTF-8"
                   schema="http://schemas.opengis.net/gml/
                           3.2.1/feature.xsd" default="true"/>
      <wps:Format mimeType="application/json"
                   encoding="UTF-8"/>
    </wps:ComplexData>
  </wps:Input>
</wps:ProcessOffering>
Synch/Asynch execute

- In WPS 1.0 asynch execute was steered by several flags
  - Impossible combinations could appear
- WPS 2.0:
  - One flag „mode“, synch, asynch or auto
  - Auto lets the server decide whether the job is run synch or asynch
<wps:Execute
    service="WPS" version="2.0.0" response="document" mode="async">

    <wps:Input id="INPUT_GEOMETRY">
    </wps:Input>

    <!-- Uses default output format -->
    <wps:Output id="BUFFERED_GEOMETRY" wps:dataTransmissionMode="reference">
    </wps:Output>

</wps:Execute>
KVP for execute

- KVP for execute was removed
  - Too complicated because of double URL encoding and nested inputs
  - Could be re-introduced as extension
StatusInfo

- JobID
- Status (Running, Succeeded, …)
- Estimated completion
- NextPoll
- Expiration date
- Percent completed
The scope for Process Profiles

- Align the **interfaces** of different implementations
- Align the **behavior** of different implementations

**the signature** (or interface) → Defines what goes in and comes out

... Refers to the process mechanics
... Describes how the outputs are derived from the inputs
... Defines the result quality
Fundamental difference: Planar vs geodesic buffering

North Korea’s missile threat

<table>
<thead>
<tr>
<th>Type</th>
<th>Maximum range</th>
<th>Payload</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodong</td>
<td>1,300 km (810 miles)</td>
<td>700 kg (1,550 pounds)</td>
<td>Currently deployed</td>
</tr>
<tr>
<td>Taepodong-1</td>
<td>Up to 10,000 km</td>
<td>Several hundred kg</td>
<td>Test failed 1998, not yet tested</td>
</tr>
<tr>
<td>Taepodong-2</td>
<td>10,000-15,000 km</td>
<td>Several hundred kg</td>
<td>Not yet tested</td>
</tr>
</tbody>
</table>

Source: Task Force for US Korea Policy, Centre for International Policy
Hierarchical profiling approach (Buffer)

- **Concept Buffer**
- **Generic Profile**
  - GeodesicBuffer
- **Implementation Profile**
  - GeodesicBuffer on GeoJSON
- **Process Implementation**
  - [http://my.process.geodesic-buffer](http://my.process.geodesic-buffer)
Conformance tests

- More tests than in version 1.0
- Still no release of ATS for 1.0
  - Difficulties testing execute
- Work is needed to create ATS for 2.0
  - Together with test developers
  - Planned to do this in the context of OWS-11
Conformance tests

- Proposal to introduce an echo process
- Offers Complex-, Literal, BBoxData
- Returns inputs unchanged
- Execute operation should be testable
Summary

- WPS 2.0:
  - Modular standard
    - Conceptual service model
    - Process model
  - Leaner specification
  - More metadata
  - Fine grained profiles
More information:

WPS 2.0 SWG
OGC TC Discuss/Announce list

b.pross@52north.org