Apresentando um Web Service de Processamento Geográficos segundo o padrão WPS da OGC

Presenting a Geographic Web Processing Service following OGC WPS Standard

Dezembro / 2009

José Roberto M. Garcia[1]

[1] *Centro de Previsão de Tempo e Estudos Climáticos (CPTEC), Instituto Nacional de Pesquisas Espaciais (INPE) – Rod. Pres. Dutra, km 40 – Cachoeira Paulista, SP – Brasil CEP: 12630-000*

Abstract *There is nowadays a sort of infraestructure mechanisms that provide spatial data to users – catalogues, web services, maps, datasets – however specialied users need advanced services that process and transform these data into usefull information with no effort. To address these issue, the Open GeoSpatial Consortium has proposed a specification called WPD to stabilish a service development standardizationnd share algoritms and funtionalities. This article presents a WPS implementation following the standard.*

Key words: *OGC, WPS, web services, geoprocessing, geographical information systems*

Resumo *Já há inúmeras infraestruturas para fornecer dados espaciais aos usuários – catálogos, web services, mapas, conjunto de dados – entretanto usuários especialistas requerem serviços mais avançados que processem estes dados transformando-os em informações úteis de modo que sejam acessados com o mínimo de esforço possível. Para resolver esta questão o Open Geospatial Constortium (OGC) propôs uma especificação chamada Web Processing Service (WPS) para padronizar a construção de tais serviços e compartilhar algorítmos e funcionalidades. Este artigo apresenta uma implementação de um serviço WPS segundo este padrão.*

Palavras-chave*: OGC, WPS, web services, geoprocessamento, sistema de informações geográficas*

Introdução

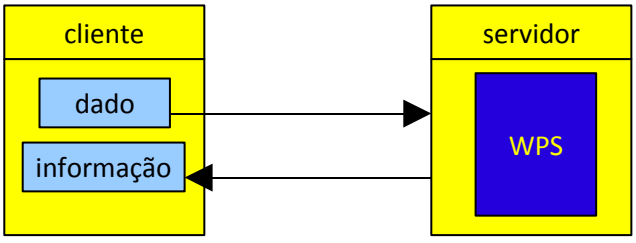
Pesquisas em Sistemas de Informação Geográficas têm sido ampliadas e melhoradas em razão do avanço tecnológico, do aprimoramento do conhecimento humano e do surgimento de novos problemas e necessidades. Além disso, um esforço cooperativo vêm sido liderado por duas instituições internacionais (ISO [1] e OGC [2]) visando definir padrões e especificações para a interoperabilidade de sistemas. Este trabalho torna possível que as instituições que visam criar uma infraestrutura de dados espaciais [3] possam compartilhar seus dados geográficos.

A OGC é uma instituição internacional de renome que lidera a criação de padrões que permitem o desenvolvimento de sistemas geoespaciais interoperáveis. Uma das mais recentes especificações padronizadas pela OGC é o *Web Processing Service* (WPS) [4]. Esta especificação define um mecanismo em que o cliente pode submeter uma tarefa que envolva processamento espacial a um servidor. Em outras palavras, esta especificação padroniza a maneira que cálculos SIG estejam disponíveis na *Web*. Neste artigo o autor relata as dificuldades e detalhes mais importantes ao implementar um WPS segundo esta especificação.

Web Preocessing Service (WPS)

O WPS [4] é uma das mais recentes especificações da OGC. Alguns trabalhos têm sido realizados para avaliar a tecnologia [5][6][7].

Este padrão define um mecanismo em que um cliente execute qualquer tipo de processamento geográfico disponível em um servidor remoto e receba o dados geoespacial resultante.



Para conseguir isso o padrão define uma série de funcionalidades que devem ser providas pelo servidor. Toda a comunicação entre cliente e servidor é realizada através do protocolo HTTP e com o uso de arquivos no padrão XML e GML (derivado de XML que tem capacidade de descrever geometrias). Caso tudo seja feito conforme a especificação, a descoberta e execução de um WPS se dá conforme a sequência a seguir:

Passo 1: o cliente envia um *request* a um servidor através de uma URL solicitando os processos existentes, por exemplo:
http://localhost:8080/wps/WebProcessingService?REQUEST=GetCapabilities&SERVICE=WPS

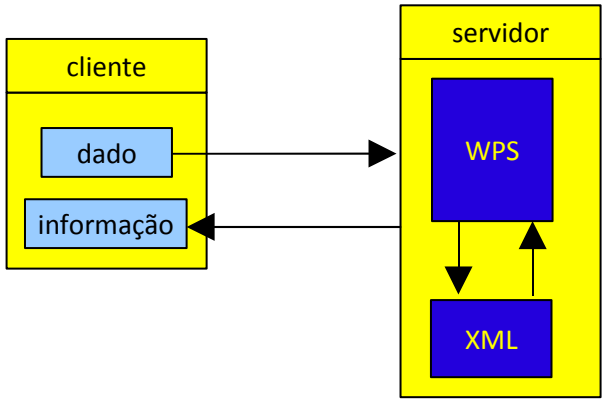
onde:

http:// → indica o protocolo que será utilizado na comunicação
localhost → a localização do servidor
8080 → a porta que ele atende no servidor
wps → nome dado ao servidor
WebProcessingService → nome dado ao serviço no servidor

A partir deste ponto da URL, iniciam-se as padronizações específicas do WPS.

REQUEST → indica que o parâmetro a seguir será o tipo de requisição que o cliente está fazendo ao servidor, no caso de WPS são 3 possíves.
GetCapabilities → informa ao servidor que o cliente deseja saber quais os serviços de processamento disponíveis nele.
SERVICE → indica que o parâmetro a seguir será o tipo de serviço que está sendo pedido na requisição.
WPS → indica que a requisição se refere a um serviço WPS

Uma vez que o servidor já está preparado para responder a esta requisição ele a interpreta, consulta sua configuração contida em arquivo XML que indica quais os serviços disponíveis e envia a resposta (*request*) ao cliente, também no formato XML.

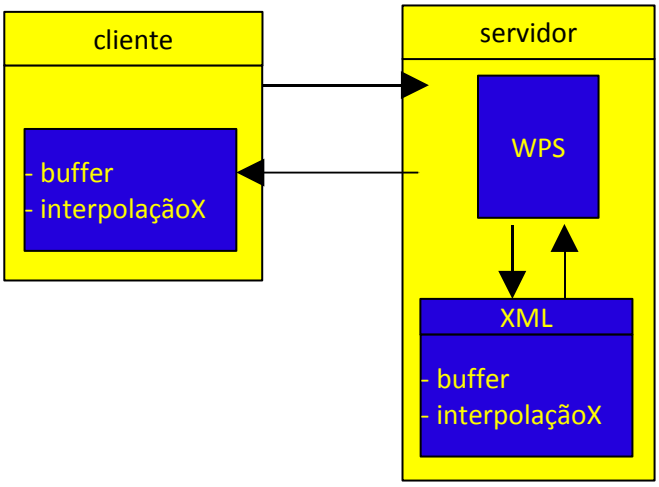


Segue abaixo uma possível resposta em XML de forma reduzida.

```
<Capabilities version="0.4.0“>
  <ows:ServiceIdentification>
  …
  </ows:ServiceIdentification>
  <ProcessOfferings>
    <Process>
    <ows:Identifier>Buffer</ows:Identifier>
    </Process>
  </ProcessOfferings>
  <ProcessOfferings>
    <Process>
    <ows:Identifier>InterpolaçãoX</ows:Identifier>
    </Process>
  </ProcessOfferings>
</Capabilities>
```

No exemplo acima o servidor respondeu, em XML padronizado pela especificação, que oferece dois tipos de processamento: **Buffer** e **InterpolaçãoX.**

O trabalho do cliente é de interpretar este XML recebido de volta e apresentar as operações disponíveis na sua interface, para que o usuário possa escolher qual operação deseja.



Passo 2: De posse das informações contendo os processos disponíveis no servidor, o cliente escolhe o serviço que ele quer ser executado e pede maiores informações sobre o serviço. Este pedido é feito através de uma URL em que o serviço requerido é especificado, como por exemplo:

http://localhost:8080/wps/WebProcessingService?REQUEST=DescribeProcess&Identifier=Buffer&SERVICE=WPS&VERSION=0.4.0

onde:

REQUEST → indica que o parâmetro a seguir será o tipo de requisição que o cliente está fazendo ao servidor, no caso de WPS são 3 possíves.
DescribeProcess → informa ao servidor que o cliente deseja saber todas as informações sobre um serviço de processamento específico como quais parâmetros de entrada e saída e tipo de dado de cada parâmetro.
Identifier → indica que o parâmetro a seguir será o nome do serviço que está sendo requerido
Buffer → informa que o nome do serviço requerido
SERVICE → indica que o parâmetro a seguir será o tipo de serviço que está sendo pedido na requisição.
WPS → informa que a requisição se refere a um serviço WPS
VERSION → Indica que o parâmetro a seguir será a versão da especificação WPS que o cliente espera a resposta.
0.4.0 → informa que o serviço foi construído de acordo com a versão 0.4.0 da especificação WPS.

Uma vez que o servidor já está preparado para responder a esta requisição ele a interpreta, consulta a configuração específica do serviço requerido contida em outro arquivo XML que contém as informações do serviço e envia a resposta (*request*) ao cliente, também no formato XML.

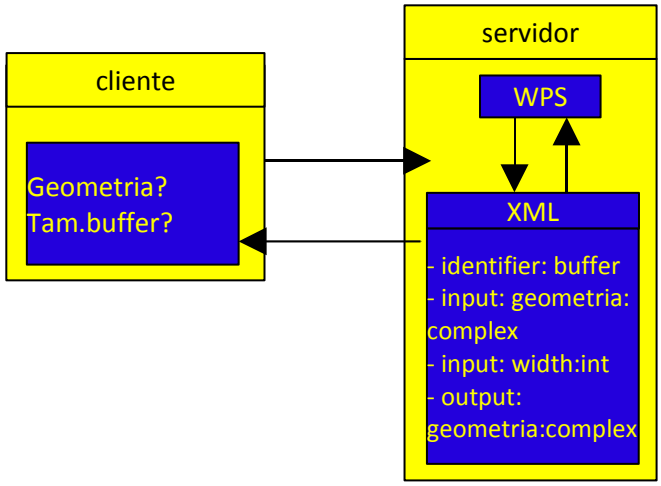
Segue abaixo uma possível resposta em XML de forma reduzida.

```
<ProcessDescriptions>
 <ProcessDescription>
  <ows:Identifier>Buffer</ows:Identifier> ...
  <DataInputs>
   <Input>
    <ows:Identifier>geometry</ows:Identifier>
    <ComplexData defaultSchema="schemaNS"/>
    <MinimumOccurs>1</MinimumOccurs>
   </Input>
   <Input>
    <ows:Identifier>width</ows:Identifier> ...
    <LiteralData>
    <ows:DataType ows:reference="xs:int"/> ...
    <ProcessOutputs>
   <Output>
    <ows:Identifier>BufferResult</ows:Identifier>
    <ComplexOutput defaultSchema="schemaNS"/> ...
</ProcessDescriptions>
```

No exemplo acima o servidor respondeu, em XML padronizado pela especificação, num formato em que visualmente podemos interpretar algumas informações, por exemplo:

- o nome do processo descrito chama-se **Buffer**
- ele espera como *input* uma geometria de um tipo de dado complexo
- espera também uma largura cujo tipo é um número inteiro.
- a resposta (*output*) será informada através do rótulo BufferResult de um tipode dado complexo.

O trabalho do cliente desta vez é de interpretar este XML recebido e preparar a interface para que o usuário possa fornecer os *inputs* necessários para disparar o processo no servidor.



Passo 3: De posse das informações contendo os inputs e output do processo escolhido, o cliente informa os dados necessários para executar o processo e envia o comando para ser executado. Este comando é feito através de uma URL em que os dados necessários são informados através de um arquivo GML, que é necessário pois os dados requeridos podem ser geometrias e os arquivos padrão GML conseguem representá-las. Segue abaixo um exemplo de como poderia ser um arquivo GML que invoca o serviço WPS.

```
<Execute>
<ows:Identifier>Buffer</ows:Identifier>
<DataInputs>
<Input>
    <ows:Identifier>geometry</ows:Identifier>
    <ComplexValue schema="schemaNS">
        $XML_Geometries
    </ComplexValue>
</Input>
<Input>
     <ows:Identifier>width</ows:Identifier>
     <LiteralValue>100</LiteralValue>
</Input>
</DataInputs>
</Execute>
```

No exemplo acima pode-se perceber que o cliente informou que quer executar o processo Buffer e esta passando como parâmetros uma geometria e um tamanho de buffer. O servidor deve processar este arquivo, reconhecer os dados passados, fazer o processamento solicitado e enviar uma resposta.

De acordo com os mecanismos de execução do processo configurados no servidor através de um arquivo XML específico do processo, o servidor executa o que está sendo solicitado e envia uma resposta também em formato GML pois é um formato que consegue representar geometrias.

Passo 4: De posse da resposta enviada pelo servidor no formato GML, o cliente interpreta o arquivo e mostra em sua interface gráfica o resultado da execução do processo.

```
<ExecuteResponse>
  <ows:Identifier>Buffer</ows:Identifier>
  <DataInputs>
    ...
  </DataInputs>
  <OutputDefinitions>
    ...
  </OutputDefinitions>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>BufferResult</ows:Identifier
      <ComplexValue schema ="schemaNS">
        $XML_Geometries
      </ComplexValue>
    </Output>
  </ProcessOutputs>
</ExecuteResponse>
```

Neste exemplo de resposta GML, pode-se notar que o indentificador da resposta do processo chama-se BufferResult e o tipo do resultado é um dado complexo.

E assim termina o ciclo da comunicação cliente servidor para um tipo de serviço específico.

Implantação

Tanto o servidor como o cliente podem ser desenvolvidos em qualquer linguagem de programação desde que:

- consigam tratar *requests* e *responses* HTTP.
- consigam implementar um servidor.

O servidor deve seguir as especificações definidas pela OGC para que os *requests* e os *responses* possam ser intepretados de maneira correta e padronizada.

Como exemplos de servidor pode-se citar:

- Em PYTHON → pyWPSem www.pyWPS.org
- Em JAVA → 52nWPS em http://52north.org/

O processo pode ser criado tanto na linguagem de programação do servidor como numa linguagem qualquer desde que o a linguagem em que o servidor foi desenvolvida tenha capacidade de executar processos externos. Um exemplo seria ter um processo desenvolvido em Fortran sendo executado via servidor desenvolvido em Java.

Após o processo ser desenvolvido é preciso que ele seja registrado no servidor para que ele possa ser divulgado externamente, ou seja, é preciso informar o servidor que o processo está existe. Isso é feito através de uma arquivo XML que é colocado numa área específica do sistema de arquivos do servidor.

```
<AlgorithmRepositoryList>
  <Repository name="LocalRepo"
    className="wps.server.LocalRepo">
  <Property name="Algorithm">
    wps.server.Buffer
  </Property>
  <Property name="Algorithm">
    wps.server.DouglasPeuckerAlgo
  </Property>
  <Property  name="Algorithm">
    wps.server.Intersection
  </Property>
 </Repository>
...
```

Neste exemplo, pode-se perceber que há 3 processos declarados (Buffer, DouglasPeucker e Intersection) e todos estão cadastrados dentro de uma propriedade chamada Algorithm, que serve para separar logicamente os diferentes processos a gosto do servidor.

Uma vez que o process esteja divulgado é necessário que ele seja detalhadamente descrito, para que o cliente saiba

como chamá-lo, quais os tipos de *input* são requeridos e e qual o *output* gerado. Isto é feito através de um arquivo em XML que também é colocado numa área específica do sistema de arquivos do servidor e chama-se ProcessDescription.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Este arquivo descreve o processo .... -->
<wps:ProcessDescriptions
  <ProcessDescription wps:processVersion="2"
statusSupported="true" storeSupported="true">
 <ows:Identifier>wps.server.Buffer</ows:Identifier
>
 <ows:Title>Create a buffer .... </ows:Title>
 <DataInputs>
  <Input minOccurs="1" maxOccurs="1">
   <ows:Identifier>data</ows:Identifier>
   <ows:Title>Polygon to be buffered</ows:Title>
   <ComplexData>
    <Format>text/XML</Format>
   </ComplexData>
  </Input>
  <Input minOccurs="1" maxOccurs="1">
   <ows:Identifier>width</ows:Identifier>
   <ows:Title>Buffer Distance</ows:Title>
   <LiteralData>
    <Format> xs:doubl</Format>
   </LiteralData>
  </Input>
 </DataInputs>
 <ProcessOutputs>
  <Output>
   <ows:Identifier>result</ows:Identifier>
   <ows:Title>Buffered Polygon</ows:Title>
   <ComplexOutput defaultFormat="text/XML"
   </ComplexOutput>
  </Output>
 </ProcessOutputs>
 </ProcessDescription>
</wps:ProcessDescriptions>
```

Como pode ser notado, este arquivo contém *tags* específicas para descrever o processo, os *inputs* e o *output*.

Considerações sobre o WPS

É sabido que um sistema computacional tem condições de permanecer em operação initerruptamente, portanto isso gera um grande benefício para os usuários pois os processos podem ser executados a qualquer hora de qualquer dia desde que, é claro, o sistema computacional esteja disponível.

Como a especificação WPS da OGC é baseada no protocolo HTTP para estabelecer a comunicação entre cliente e servidor, isso a torna independente de plataforma, ou seja, não importa qual o ambiente computacional os clientes e servidores são desenvolvidos, uma vez que a utilização é o HTTP, a formatação dos *requests* e *reponses* já está definida e qualquer ambiente consegue interpretá-la.

Um outro aspecto relevante no uso de WPS é que podemos processar dados geográficos onde quer que eles estejam, seja em repositórios locais ou remotos. Para que isso seja possível o cliente deve ter também o serviço WFS implementado. Com ele, conseguimos buscar o dado propriamente dito e trabalhar com ele.

Dificuldades

A especificação WPS é bastante nova (primeira versão foi divulgada em 2005) e portanto ela não está consagrada na comunidade geoespacial.

Em face disso, há pouco material didático disponível, principalmente na última versão, o que dificulta qualquer implementação. O que existe de concreto mesmo é a especificação propriamente dita, tornando qualquer trabalho em que o foco é ter resultados rápidos para avaliar o funcionamento numa tarefa não muito fácel de cumprir.

A documentação didática disponível é imprecisa, pois descreve os passos de implementação de forma errada, contém incompatibilidade de *software* pois os *softwares* envolvidos na documentação já estão em versões mais atuais e, além disso, há muita dependência de *softwares* remotos e qualquer atualização demanda muito tempo de espera por parte do implementador.

Conclusão

O reuso das rotinas é um fator muito relevante para apostamos na concretização da especificação. Com ela, pode-se executar procedimentos desenvolvidos por intituições de alta credibilidade, eliminando a necessidade de desenvolvimentos *ad-hocs*.

Um outro aspecto bastante importante é o fato de podermos realizar o processamento “em lote”, ou seja, encadearmos vários processamentos em um pacote só, como se fosse um processamento único. Este mecanismo aliado a possibilidade de podermos programar o processamento para ser disparado a qualquer tempo e intervalo cria uma ferramenta ímpar a execução de tarefas de uso rotineiro.

O autor deste artigo aposta na consagração da especificação WPS da OGC em vista de que os aspectos acima descritos são desenvolvidos sob o alicerce de padrões consagrados não só pela OGC como da W3C.

Referências

1. ISO/IEC: Geographic Information – Reference Model. International Standard 19101, ISO/IEC (2002)
2. Open GIS Consortium, Inc.: OpenGIS Reference Model. OpenGIS Project Document 03-040, Open GIS Consortium, Inc. (2003)
3. Global Spatial Data Infraestructure Association: online. Acesso Dez 2009 em http://www.gsdi.org
4. Open GIS Consortium, Inc.: OpenGIS Web Processing Service Implementation Specification. OpenGIS Standard 05-007r7, Open GIS Consortium, Inc. (2007)
5. Michaelis, C.D., Ames, D.P.: Evaluation and implementation of the ogc web processing service for use in client-side gis. Geoinformatica (2008)
6. Stollberg, B. & Zipf, A. (2009): Development of a WPS Process Chaining Tool and Application in a Disaster Management Use Case for Urban Areas. UDMS 2009. 27th Urban Data Management Symposium, Ljubljana , Slovenia.
7. Carlos Granell, Laura Díaz, Michael Gould, Victor Pascual, Jordi Guimet, Paola Carrara, Monica Pepe. Developing geoprocessing services for a hydrological model application. In Proceedings of 27th EARSeL Symposium: Geoinformation in Europe (EARSeL 2007). Bolzano (Italy), June 2007.