NOUMEA : A Model-Driven Framework for WPS Development

Jean-Philippe Babau
Lab-STICC, UBO, Brest, France
jean-philippe.baba@univ-brest.fr

Keywords. WPS, MDE, framework

1 Introduction

In an open data world, the Open GIS Consortium (OGC) standards allow the geoscience community to share data (WMS: Web Map Service and WFS: Web Feature Service) and processes (WPS: Web Processing Service) using standard web services. The WPS [1] defines a process model through a list of descriptors, a list of inputs (requests) and a list of outputs (responses). The standard also defines how a client can request the execution of a process and how the output of the process is handled.

As mentioned in the standard, a WPS process is an atomic brick that performs a specific geospatial calculation. WPS composition enables the creation of complex and repeatable WPS workflows. In the literature (see [2, 3, 4]), four WSO (Web Services Orchestration) approaches are proposed:

- WSO standards (i.e WSDL, SOAP and BPEL), are used to orchestrate WPS. This solution is a classic solution for orchestrating web services. But as mentioned in [2], a first problem with this approach is that not all orchestration engines support the WPS protocol invocation (HTTP POST and / or GET). The second problem is that orchestration engines cannot handle raw binary data (provided by WMS invocations).
- A specific WPS plays in a centralized architecture an orchestration role, activating the other WPS. As an alternative, WPS are chained in a cascading architecture to achieve a specific task, WPS are dependent on each other. These two approaches are generally used, although performance may be degraded due to many data exchanges.
- Simple WPS functionalities are implemented as class functions and the composite WPS is implemented by calling directly the class functions. The advantage of this solution is the performance, data is exchanged locally. But it is a centralized and code-centric approach. Local functions are not exported as WPS and cannot be reused in a distributed architecture.

To develop efficient and reusable complex WPS, a key challenge for scientists is to choose the adapted approach. And, due to the technical complexity of WSO, scientists need an easy-to-use development environment, as proposed by RichWPS [5].

In this paper, we propose a Model-Driven Engineering approach [6] to facilitate the development of efficient, reusable and complex WPS. The proposed NOUMEA framework is based on an Ecore reference model (Eclipse Modeling Framework (EMF) [7]). From this model, using model transformations, it is possible to generate code and to combine the three last approaches. As usual with MDE, technical aspects are hidden to the WPS designer. The approach can be used by scientists to easily describe a WPS workflow using a graphical interface (as with the WPS Builder tool [8]). The WPS code, including local or remote WPS calls is generated. The generated code facilitates WPS testing and WPS deployment. During generation, controls are performed to ensure the workflow correctness. Using the NOUMEA framework, the scientist can focus on the domain-specific part of the WPS.

2 NOUMEA framework

Based on the OGC standard, a WPS workflow meta-model, called wfwps, is implemented as an Ecore model. This model is close to the UML model provided as an informative model by the standard. The meta-model represents the concepts used to describe a workflow, the WPS, and the information needed to drive workflow verification and code generation.

A WPS is classically described by a set of EString attributes (identifier, title and abstract) and a list of Inputs and a list of Outputs. A WPS may be a LocalWPS (implemented by a function) or a Re-
moteWPS (defined by an extern Web Service). The latter is characterized by its url and the WPS versionNumber. Each Input and Output is described by a set of EString attributes (identifier, title and abstract) and a mandatory type (Float, Integer, String, Geometry, Raster). In the proposed model, the data type declaration is mandatory (optional for the standard) to perform type checking. An Input may be Optional and is characterized by a DefaultValue and by the minimal and maximal number of occurrences.

A Workflow is an acyclic graph of WPS. Edges of the graph are called Links. A Link refers to one Output and to one Input. Unrelated inputs, resp. outputs, are the inputs, resp. the outputs, of the composite WPS. In a Workflow, an Input of a WPS may be linked to a ReferencedData obtained by executing a WMS (to get a Raster) or a WFS (to get a Geometry encoded as a FeatureCollection). The ReferencedData is characterized by an url and a layer attribute.

For generation purpose, each WPS and each Workflow is associated to a RepositoryName and a ClassName. If they are not defined, the code generator computes by-default values.

From a wfwps model, we propose a set of verification rules. We give here a non-exhaustive list of them:

- the inputs of a WPS have different identifiers;
- in a Link, the type of the output is compatible with the type of the input;
- in a Workflow, an Output cannot be linked to an Input of the same WPS;
- a Workflow an implicit cycle; a Workflow is an acyclic graph of WPS;

From a correct model, we propose a model-based Java code generator that generates:

- for each LocalWPS, the specific Java code to declare a WPS in its dedicated class. All the declarations and specific WPS Java annotations are generated; the behavior of the WPS function is the call of the function specified in the model.
- for each workflow, a WPS implemented through a sequence of WPS invocations (java function call for LocalWPS and Web Service call for RemoteWPS); the sequence respects the pre-order specified by the WPS graph; the unrelated optional inputs are initialized with the by default values; the referencedData are initialized by the corresponding WMS or WFS call; type cast is added if necessary (Integer->Float, Float->Integer).

From the model, it is also possible to generate a WPS testing environment for both client and server sides. The generated WPS can be then easily integrated in a Web Service server. We use Geoserver [9] that is an open source server for sharing geospatial data and services. The framework development is still in-progress and experimented for the development of a set of complex WPS dedicated to coastline evolution analysis [10].

3 References

10. Suanez S., Cariolot J.-M., Cancouët R., Ardhuin F., Delacourt C. ÒDune recovery after storm erosion on a high-energy beach: Vougot beach, Brittany (France)Ó Geomorphology, 139-140, 16-33, 2011.