NOUMEA : A Model-Driven Framework for WPS Development

Jean-Philippe Babau, Lab-STICC / UBO
Mathias Rouan, LETG / IUEM

Master students : Yasmin Ait Maksene, Yanis Remila, Fatah M'Sili
OGC standards

• OGC Web Services
  • Web Map Service
    • Request georeferenced maps on the web
  • Web Feature Service
    • Request geographical objects on the web
  • Web Processing Service
    • Geospatial processing services

• Standardization
  • Service description
  • Inputs and Outputs description
  • Discovery services

• A standard to share geographical data and processes
Case study

- Coast Line evolution analysis
  - A set of LineString

- A reference line
  - Orthogonal radials (length, interval, direction)

- A coast line evolution
  - Intersections with radials
  - Distances with the reference line
• Atomic WPS
  • A WPS is a function implemented in a programming language (Java, Python, …)
  • Problem: how to program and deploy a function as a WPS?

• Workflow of WPS
  • Adapted to develop complex services, reusing correct and efficient existing WPS
  • Problem: how to design a safe and efficient workflow?

• Challenges
  • Tools to automate deployment of functions
  • High-level description of workflows
  • Verifications
WPS workflow implementation

• WSO
  • High-level description
  • Not all WSO integrate OGC WPS protocols
  • No raw binary data (WMS) integration
  • Possible performance degradation due to data exchanges

• Code-centric approach
  • An atomic WPS = a function
  • A workflow = a function
  • Efficient approach due to local processing
  • Difficult to build / integrate reusable WPS

• A WPS plays the role of the workflow scheduler
  • Reuse existing WPS
  • WPS-workflow programming
  • Possible performance degradation due to data exchanges

• Solution
  • A high-level description of the workflow
    • Edition, verification, optimization
    • Code generation
  • Mixing local and remote calls
NOUMEA modeling

• Model-Driven approach
  • A model to hide technological details
  • A model plays a central role for
    • Verification
    • Code generation
    • User interaction

• Eclipse Modeling Framework
  • Object-oriented models
  • Ecore models: class diagram with OCL constraints

• WPS
  • Implementing the standard as an Ecore model

• Workflow
  • An Ecore model of chained services
    • Including local and remote WPS calls
    • Including WMS and WFS calls
NOUMEA WPS modeling
NOUMEA verification

• OCL constraints
  • Inputs (Outputs) of a WPS have different names
  • Inputs (except optional) and outputs are connected
  • Same type for input and output of a link

Invariant sameType:

\[(\text{input.oclIsTypeOf(OutputReference) and output.oclIsTypeOf(InputReference)})\]

implies \(\text{input.oclAsType(OutputReference).output.type = output.oclAsType(InputReference).input.type}\);

• Java implemented verification
  • No cycle
  • No recursion
NOUMEA model edition

- Graphical visualization of models
  - Sirius technology
  - Easy to design workflows
NOUMEA code generation

- Hide technological details
  - Java implementation
- Based on templates
  - Acceleo technology

Java code

```java
package Tests;

public class HW {
    public String helloWorld(String world) {
        return("Hello "+world);
    }
}
```
package TestWPS;

import org.geotools.process.factory.*;
import org.geotools.text.Text;

public class HelloWorld_class extends StaticMethodsProcessFactory<HelloWorld_class> {

    public HelloWorld_class() {
        super(Text.text("WPS"), "TestWPS", HelloWorld_class.class);
    }

    @DescribeProcess(title="HelloWorld", description="WPS")
    @DescribeResult(name="result", description="input")
    public static String HelloWorld(@DescribeParameter(name="input", description="input") String world) {
        return ("Hello "+world);
    }
}
Model-driven approach

WPS modelling

Generated from source
Generated from source
Generated from source
Generated from source

Local WPS HelloWorld

- Name: HelloWorld
- Abstract: WPS
- Package Name: Tests
- Class Name: HW
- Function Name: helloWorld
package TestWPS.WPSpackage;

import org.geotools.process.factory.*;
import org.geotools.text.Text;
import Tests.HW;

public class HelloWorld_class extends StaticMethodsProcessFactory<HelloWorld_class> {

    protected static HW callObject;

    public HelloWorld_class() {
        super(Text.text("WPS"),"TestWPS",HelloWorld_class.class);
        callObject = new HW();
    }

    @DescribeProcess(title="HelloWorld",description="WPS")
    @DescribeResult(name="result",description="input")
    public static String HelloWorld(@DescribeParameter(name="input",description=" input") String input) {
        String result;
        result = callObject.helloWorld( input);

        return result;
    }
}
NOUMEA User Interface

- Configuration
- Project folders

![NOUMEA User Interface](image-url)
NOUMEA User Interface

- Modeling
  - Atomic WPS: automatic generation of the model from the code
• Modeling
  • Remote WPS: automatic generation of the model from the WPS description
NOUMEA User Interface

• Modeling
  • Workflow: graphical editor
NOUMEA tooling

- Code generation
  - Java code generation

![Noumea User Interface](image-url)
NOUMEA tooling

- Deployment
  - Automatic deployment on GeoServer
Case study

- A workflow of 3 atomic WPS
  - Reusable WPS

- Easy to develop and to deploy
  - Concentrate on specific code
Case study

- A specific workflow
  - Data are provided by WFS calls

- Easy to develop and to deploy
  - Reuse of workflows
**Conclusion**

- A model-based framework for OGC WPS development and deployment
  - Java function deployed as WPS
  - Graphical design and java code generation for workflows
  - Workflow verification

- Future works
  - Multiple outputs
  - Other implementation languages
  - Geometry typing (modeling and code generation)
  - Exception management (code generation)
  - Performance and security
  - Software Engineering principles to define architectural styles
Questions ?

https://github.com/jpbabau/Noumea

babau@univ-brest.fr