BDD, ISO 19105 and INSPIRE NS Conformance

Francisco J. Lopez-Pellicer, Javier Lacasta – IAAA, Universidad Zaragoza (SPAIN)
Jesús Barrera, José M. Agudo – GeoSLab (SPAIN)
Paloma Abad Power, Alejandra Sánchez Maganto, Emilio López Romero – CNIG (SPAIN)

INSPIRE Conference 2014, Aalborg, Denmark
19 June 2014
Outline

1. Testing Geographic Information software is hard
   • And testing INSPIRE NS conformance is even harder

2. Introduction to Behaviour Driven Development (BDD)
   • Or how to promote common understanding during tests

3. BDD vs ISO 19105:2000 testing methodology
   • Can map BDD to ISO 19105?

1. A BDD-based compliance testing tool for INSPIRE NS
   • Current status & future steps
Testing GI software is hard

- Authors say software testing is ...
  - Ad hoc, expensive and unpredictably effective [1]
  - Costly and risky especially when Web services are involved [2]

- GI has inherent complexities
  - Complex models
  - Distributed systems
  - Many application scenarios
  - ...

And testing INSPIRE NS is even harder

1. Many stakeholders
   - Out of GI domain: non-GIS developers
   - Managers, domain experts, users: lack of technical skills

1. Current tools require very skilled people
   - Schematron (e.g. JRC – old version)
   - UML models (e.g. JRC – new version)
   - OGC CTL + XSLT (e.g. GDI-DE)

2. Cultural barrier
   - Generated test reports expressed in technical parlance
   - How to make intelligible a non-conformity to stakeholders?
Example of Schematron rule (excerpt)

```xml
<sch:rule
context="//gmd:identificationInfo/*/gmd:citation/gmd:CI_Citation">
  <!-- Title -->
  <sch:let name="noResourceTitle" value="not(gmd:title) or gmd:title/@gco:nilReason='missing'"/>
  <sch:let name="resourceTitle" value="gmd:title/*/text()"/>
  <sch:assert test="not($noResourceTitle)">
    <sch:value-of select="$loc/strings/alert.M35/div"/>
  </sch:assert>
  <sch:report test="not($noResourceTitle)">
    <sch:value-of select="$loc/strings/report.M35/div"/>
    <sch:value-of select="$resourceTitle"/>
  </sch:report>
</sch:rule>
```

Communication between involved & affected

AFFECT/AFFECTED (Stakeholders)

INVOLVED

Service Developer
Developer (e.g. GIS Java developer)
Database Admin. (e.g. Oracle BDA)
System Admin. (e.g. Network admin.)

Service Owner
Manager (e.g. Project Manager)

Service Consumer
End User (e.g. Farmer)
Domain Expert (e.g. Remote Sensing expert)

Secondary Stakeholder
Government bodies (e.g. EEA, Eurostat)
Research centers (e.g. JRC)
Communication between involved & affected

Use an ubiquitous language for expressing what is going to be tested

Ubiquitous language

• Language built to be shared by developers and users
  • Common, rigorous
  • Promote common understanding

Behaviour Driven Development

• **BDD** [4]
  - Lightweight and non-formal Model-Based Testing [5] software development process based on an ubiquitous language

• **Involve affected in testing!**
  - Software developers and stakeholders collaborate in developing human readable Abstract Test Suites (ATS) for acceptance tests
  - ATS are written in Gherkin (ubiquitous language)
    - https://github.com/cucumber/cucumber/wiki/Gherkin

• **ATS are runnable in BDD**
  - Lot of tools: RSpec, JBehave, StoryQ, SpecFlow …
  - Test reports expressed in ATS terms!!!

Runnable ATS

- Gherkin spec
  - Human readable ATS
  - Few but key conventions

- Step definitions
  - “Real” testing code
  - Reusable/Parametrizable
  - Runtime binding based on conventions

- Execution tool
  - Run tests and create reports based on ATS
Gherkin

• **Stub of a human readable ATS →**
  • In Gherkin

• **And the testing code?**
  • Developers use BDD tools to produce automatically testing code stubs in several languages
  • Isolate ATS from Executable Testing Suite (ETS) implementation details

---

**Feature** [title]

In order to [some benefit] as [role] I want [some feature]

**Scenario** [title]

*Given* [context]*

*When* [event]*

*Then* [outcome to test]
Feature: Requirement 50
The mandatory VERSION parameter. The value "1.3.0" shall be used for GetMap requests that comply with the ISO 19128 standard.

Scenario: Check if version Parameter with value to 1.3.0 is accepted and that an exception is thrown when no parameter version is used

Given the service's capabilities document

Then a request with VERSION=1.3.0 parameter should return an image

And a request with no VERSION parameter should return an exception
**Gherkin (in test report)**

**Feature:** Requirement 50
The mandatory VERSION parameter. The value "1.3.0" shall be used for GetMap requests that comply with the ISO 19128 standard.

**Scenario:** Check if version Parameter with value to 1.3.0 is accepted and that an exception is thrown when no parameter version is used

**Given** the service's capabilities document [OK]

**Then** a request with VERSION=1.3.0 parameter should return an image [OK]

**And** a request with no VERSION parameter should return an exception [FAIL]
(technical details are available for devs.)
BDD and ISO 19105:2000 testing methodology

- **Mapping ISO 19105 → BDD**
  - ATS → Feature suite
  - ETS → Scenario + step definitions (adaptor code)

- **Strongest point**
  - Executable ATS are readable, semi-formal

- **Partial support (current implementations)**
  - Hierarchical ATS, conformance levels and dependences between tests

- **No support (current implementations)**
  - Conditional requirements & inconclusive verdicts
BDD and ISO 19105:2000 testing methodology

1. Selection of requirements
2. Production of ATS
3. Production of ETS
4. Execution of ETS against an IUT
5. Analysis of results
BDD applied to INSPIRE NS, why?

• **High level language**
  - The use of a human-readable language may help a better understanding of the specs by stakeholders
  - **Advantage 1:** Schematron, XSLT based approaches are low level languages and focused to developers

• **Developer friendly**
  - IDEs (Eclipse, NetBeans, VisualStudio) provide extensive support for running and debugging BDD specs
  - **Advantage 2:** CITE/CTL lacks such support because it is a niche tool
A compliance testing tool for INSPIRE NS

• **Multilingual online tool**
  - [http://idee.unizar.es/validator/?lang=en](http://idee.unizar.es/validator/?lang=en)
  - Link to the capabilities XML doc

• **Test coverage**
  - TG INSPIRE View Services
  - TG INSPIRE Discovery Services

• **Technologies**
  - Grails frontend
  - Cucumber-JVM + Gherkin for the BDD backend
    • Patched for adding missing ISO 19105 features
## Test suite

### Compliance Report

**INSPIRE Profile of WMS 1.3.0**

**Fail (2014-02-13 17:50:27 CET)**

**Justification:**

Requirements marked as 'Check' after a validation need to be checked by the user. This can be because they cannot be checked automatically or because they haven't been implemented yet.

### Set of Tests:

<table>
<thead>
<tr>
<th>Test Purpose</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1. An INSPIRE View Service shall implement the minimal mandatory behaviour from an ISO 19128 service, extended with the extensions required by the INSPIRE Directive and the Implementing Rules for View services.</td>
<td>Fail</td>
</tr>
<tr>
<td>Requirement 2. The use of ISO 19128 de jure standard as a basis for implementing an INSPIRE View service means that this service shall comply with the &quot;basic WMS&quot; conformance class as defined in this de jure standard.</td>
<td>Pass</td>
</tr>
<tr>
<td>Requirement 3. The following ISO 19128 operations shall be implemented for an INSPIRE View service: GetCapabilities, GetMap.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
# Testing a requirement

## Set of Tests: INSPIRE Profile of WMS 1.3.0

<table>
<thead>
<tr>
<th>Requirement 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Analyzed</td>
</tr>
<tr>
<td>Test Purpose</td>
</tr>
</tbody>
</table>
| Test Method | **Scenario:** If Scenario 2 has been selected, check if the `inspire_common:Conformity` element has been correctly used.  
1. Given the service's capabilities document  
2. And prefix wms is http://www.opengis.net/wms  
3. And prefix inspire_vs is http://inspire.ec.europa.eu/schemas/inspire_vs/1.0  
4. And prefix inspire_common is http://inspire.ec.europa.eu/schemas/common/1.0  
5. When there is not an `inspire_common:MetadataUrl` node in the `inspire_vs:ExtendedCapabilities` section  
**Scenario not applicable detected in 00:00:004**  
Node [`inspire_vs:ExtendedCapabilities`][inspire_common:MetadataUrl] exists  
6. Then there is a `inspire_common:Conformity` node in the `inspire_vs:ExtendedCapabilities` section |
| Verdict | Pass (2014-02-13 17:50:19 CET) |
## Current status

- **Testing artefacts produced**

<table>
<thead>
<tr>
<th>Artifact</th>
<th>View</th>
<th>Discovery</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATS Requirments Automated</td>
<td>60</td>
<td>25</td>
<td>85</td>
</tr>
<tr>
<td>Human Verification Required</td>
<td>13</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>Scenarios</td>
<td>61</td>
<td>23</td>
<td>84</td>
</tr>
<tr>
<td>Given/When/Then</td>
<td>270</td>
<td>158</td>
<td>428</td>
</tr>
<tr>
<td>ETS Methods implemented</td>
<td></td>
<td></td>
<td>72</td>
</tr>
</tbody>
</table>

- **Manual conformance testing still required!!!
- Developers couldn’t automatize 20 requirements**
Future steps

• **User feedback and review of ATSSs**
  - Implement pending requirements
  - Improving reporting

• **Support more TGs**

• **Analyze ATSSs for assessment of TGs**
  - Deriving state machines for ideal INSPIRE NS
  - Redundancies and inconsistencies in TGs
Feedback, comments are welcome!

Thanks!

Contact: Francisco J Lopez-Pellicer <fjlopez@unizar.es>