Using Processing Descriptions for Service based Schema Transformations

Luise Hutka
Stephan Mäs
Lars Bernard
INSPIRE Schema Transformation NS

Status:

- INSPIRE Regulation defines the functional requirements of an INSPIRE Transformation Network Service
- several potential architectures of Transformation NS have been proposed, some are computationally intractable
- Technical Guidance Document from 2010 specifies the service interface
  - Rule Interchange Format (RIF) is proposed for the specification of transformation rules
  - source schema descriptions as GML application schemas
- still many open questions (performance, use of query transformations, ...)

INSPIRE Conference 2011
Edinburgh
Rule Interchange Format

- W3C Recommendation from 22\textsuperscript{nd} June 2010
- exchange format for (logical) rules in the Semantic Web
- porting rules between systems with different rule encodings
- independent standard
- high expressiveness
- little implementations (so far)
- formal mathematical notations → Relatively complex
Research Goals

- use of WPS in combination with a geoprocessing language for service based schema transformations
- evaluation of the expressiveness and usability with regard to a INSPIRE use case

- approach:
  - use of the GearScape Geoprocessing Language (GGL)
  - schema transformation between
    - Source: protected sites of Saxony, Germany (Saxon State Office for Environment, Agriculture and Geology)
    - Target: INSPIRE data specification on protected sites
GGL - GearScape Geoprocessing Language

- part of the Spanish open source project GearScape
- integrated in 52ºNorth WPS implementation
- originally designed for description, transfer and execution of simple geodata analysis and processing
- based on Spatial SQL
- extensible through Java scripts
- relatively simple
- generic process description with abstract script parameters

Cortés and Leduc (2010): GGL: A geo-processing definition language that enhance spatial SQL with parameterization. AGILE 2010
Architecture

Transformation Service (WPS including GGL-Scripts)
Schema Transformation Levels

1. Renaming of classes and attributes
2. Simple attribute derivation
3. Aggregating input records
4. Complex derivation and dynamic type selection
   - logical operators for attribute derivation
5. Deriving values based on multiple features
   - e.g. spatial analysis between the features of multiple classes
6. Conflation and model generalization
   - e.g. multiple target features from one source feature

Level 1: Renaming of classes and attributes

- **GGL-Script:**

```
1 -- Rename table to "ProtectedSite" and area attribute to "officialSiteArea"
2 DOC('Source of table and area attribute to rename');
3 DECLARE source:TABLE(area:DOUBLE);
4 -- End of parameter section.
5 CREATE TABLE "ProtectedSite" AS SELECT *{EXCEPT area}, area AS "officialSiteArea"
   FROM source;
```
Level 1: Renaming of classes and attributes - RIF

```
Document

Payload
  Group
    Sentence
      If
        Exists
          ( ?ax_flurstueck-instance # source:AX_Flurstueck
            ?ax_flurstueck-instance
            [ source:amtlicheFlaechen -> ?ax_flurstuecktype- amtlicheFlaechen
            ]
          )
      Then Do
        Assert
          ( ?cadastralparcel-instance New()
            ?cadastralparcel-instance # target:CadastralParcel
            ?cadastralparcel-instance
            [ target:areaValue -> ?ax_flurstuecktype- amtlicheFlaechen
            ]
          )
    

```

schema transformation German ALK „Flurstücke“ to INSPIRE cadastral parcels
Level 2: Simple attribute derivation

<table>
<thead>
<tr>
<th>Schutzgebiet</th>
<th>ProtectedSite</th>
</tr>
</thead>
<tbody>
<tr>
<td>area {in m^2}</td>
<td>officialSiteArea {in ha}</td>
</tr>
<tr>
<td></td>
<td>siteProtectionClassification {&quot;landscape'}</td>
</tr>
</tbody>
</table>

- **GGL-Script:**

```gsql
1 -- Convert area from square meter to hectare and set default value for protection classification attribute
2 DOC('Source to transform unit of area');
3 DECLARE source:TABLE(area:DOUBLE);
4 -- End of parameter section.
5 CREATE TABLE "ProtectedSite" AS SELECT *{EXCEPT area}, (area / 10000) AS "officialSiteArea", 'landscape' AS "siteProtectionClassification" FROM source;
```
Level 3: Aggregating input records

```
1 -- Aggregate subareas based on identifier and also aggregate some thematic
    attributes
2 DOC('Table to aggregate');
3 DECLARE source:TABLE(geom:GEOMETRY(MULTIPOLYGON OR POLYGON), identifier:LONG, area:
    DOUBLE, name:STRING);
4 -- End of parameter section.
5 CREATE TABLE "ProtectedSite" AS SELECT GeomUnion(geom) AS "geometry", Sum(area) AS "
    officialSiteArea", ToString(identifier) AS "localId", ArgMin(name, 0) AS "
    siteName" FROM source GROUP BY identifier;
```
Level 4: Complex derivation and dynamic type selection

- GGL-Script:

```sql
-- Get all official protected sites with an area of at least 1,000,000 square meters
DOC('Source and required fields to test');
DECLARE source:TABLE(status:STRING, area:DOUBLE);
-- End of parameter section.
CREATE TABLE "ProtectedSite" AS SELECT *(EXCEPT area), area AS "officialsiteArea"
    FROM source WHERE (status NOT LIKE 'ES') AND (area >= 1000000);
```

ES = "einstweilig sichergestellt"
Level 5: Deriving values based on multiple features

```
1  -- Return all protected sites that are located completely in another one
2  DOC('Source to check');
3  DECLARE source:TABLE(geom:GEOMETRY);
4  -- End of parameter section.
5  CREATE TABLE "ProtectedSite" AS SELECT s1.* FROM source s1, source s2 WHERE (Within
   (s1.geom, s2.geom) = TRUE) AND (s1.geom != s2.geom);
```
Level 6: Conflation and model generalization

---

```
1  -- Create separate features representing polygons and boundary lines from one single
    protected sites feature
2  DOC('Source to create polygons and boundary lines from');
3  DECLARE source:TABLE(geom:GEOMETRY(POLYGON OR MULTIPOLYGON));
4  -- End of parameter section.
5  CREATE TABLE "ProtectedSite" AS SELECT * from source;
6  CREATE TABLE "Boundary" AS SELECT Boundary(geom), *(EXCEPT geom) FROM source;
```
Evaluation

- transformation of German protected sites data to INSPIRE data schema worked without bigger problems

- architecture:
  - client applications are relatively simple to implement, but require WPS support
  - no query transformation necessary

- GGL for schema transformation
  - supports all 6 mapping levels
  - applicable for simple mappings with schemas of little complexity
  - GGL2 (Beta) will be a completely redesigned language, much more expressive and with higher complexity

- Geoprocessing Languages and WPS can be used for schema transformation services