Reusable SHACL Constraint Components for Validating Geospatial Linked Data

Christophe Debruyne

Smals Research

Belgium

Kris McGlinn

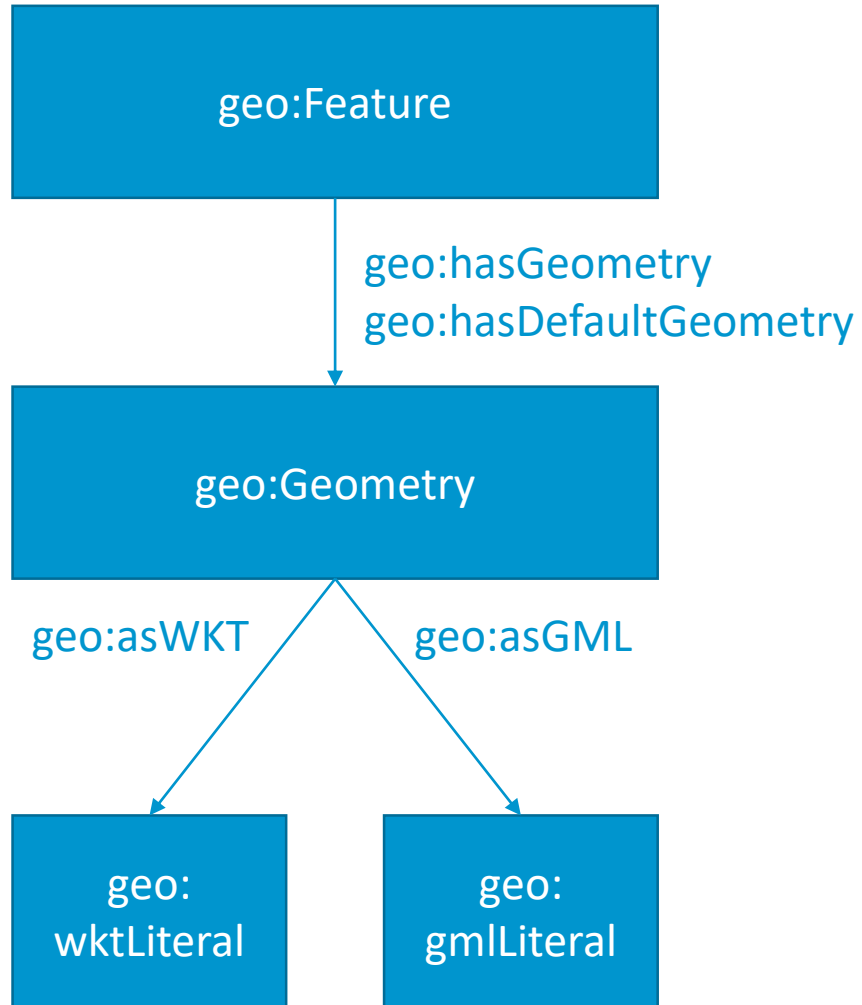ADAPT Centre, Trinity College Dublin

Ireland

2021-07-06 @ the 4th International Workshop on Geospatial Linked Data (GeoLD 2021)

Problem

- SHACL is a W3C Recommendation for validating RDF graphs. While the constructs SHACL provides may, arguably, be limited, SHACL none the less allows one to create custom components based on SPARQL. But,…

- Problem: many constraints are applicable to many applications, datasets, etc. Are people <u>continuously reinventing the wheel</u> when writing SHACL?

- As many Linked Data datasets have a geospatial dimension and GeoSPARQL is an important standard, we propose GeoSHACL; a set of constraint components for GeoSPARQL published as per Linked Data best practices.

- To the best of our knowledge, no prior work on SHACL constraints shared per Linked Data principles.

Smals
ICT for society

GeoSPARQL

geo:Feature

|
geo:hasGeometry
geo:hasDefaultGeometry
↓

geo:Geometry

geo:asWKT   geo:asGML

geo:
wktLiteral   geo:
gmlLiteral

- An OGC Standard since 2012

- GeoSPARQL provides functions (e.g., geof:sfDisjoint) and relations (e.g., geo:sfDisjoint)

- GeoSPARQL provides query-transformation rules

  E.g,. a triplepattern using geo:sfDisjoint will be rewritten as that triplepattern and the union of other graph patterns using the function geof:sfDisjoint.

- Query-transformation rules are part of the specification, but not all implementation support those (by default)

Smals
ICT for society

GeoSHACL: Design and Implementation

- Starting from GeoSPARQL's simple feature relation family providing 8 topological relations
  - equals, disjoint, intersects, touches, crosses, within, contains, and overlaps

- In GeoSPARQL, points can never be equal (as they have empty boundaries). We therefore introduced an "intuitive equals" based on contains and within.

- The behavior of the constraint components should resemble those of SHACL core comparison operators
  - Compare the lexical representation of a geometry (via a path) with either a constant or the lexical representation of a geometry via a predicate

- We will not assume that query transformation rules have been enabled. I.e., the implementation "depends" on the spatial functions applied on literals.

The availability and correctness of the functions are a different concern. GeoSHACL is not a GeoSPARQL benchmark or test suite.

Smals
ICT for society

GeoSHACL: Design and Implementation

```
# Are things that Foo contains actually within Foo?
ex:FooShape
  a sh:NodeShape ;
  sh:targetNode ex:Foo ;
  sh:property [
    sh:path (geo:sfContains geo:hasGeometry geo:asWKT) ;
    geosh:within geo:asWKT ;
  ] ;
.
```

From ex:Foo, we refer to the literals we want to compare.

Check these values whether they are within the literal of ex:Foo's geo:asWKT

Note: the domain and range of topological relations is geo:SpatialObject (geo:Feature union geo:Geometry)

```
# Implementation of geof:sfWithin constraints
geosh:withinConstraint
    a sh:ConstraintComponent ;
    sh:parameter [ sh:path geosh:within ; ] ;
    sh:validator [
        a sh:SPARQLAskValidator ;
        sh:message "Value is not within {$within}." ;
        sh:ask """
            PREFIX geo: <http://www.opengis.net/ont/geosparql#>
            PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
            ASK {
                { FILTER( geof:sfWithin($value, $within) ) }   # when comparing constants
                UNION {
                    FILTER( isIRI($within) )                   # when comparing value of property
                    $this $within ?otherValue .
                    FILTER( geof:sfWithin($value, ?otherValue) )
                }
            }
        """
    ]
.
```

Published as Linked Data

While not an ontology in the traditional sense, we have published GeoSHACL according to best practices in Linked Data and, as CC BY 4.0, and accessible via a persistent identifier.

geosh: https://w3id.org/geoshacl#

Discussion

- Little to no prior work (to the best of our knowledge)

- Our approach does not only support WKT. Support for other literals, as is the case with Apache Jena, depends on the endpoint

- When GeoSPARQL functions are not supported, the constraint components fail "gracefully". See SPARQL 1.1 Query Language - Expressions and Testing Values.

- Testing whether an endpoint supports GeoSPARQL functions is not *and should not be* GeoSHACL's concern.

  - Benchmarks do exist (*)

  - One _can_ write SHACL constraints to tests the existence / correct implementation of GeoSPARQL, _but_ then we are creating a test suite or benchmark. This may be useful to be declared separately?

(*) Jovanovik, M., Homburg, T., Spasić, M.: Software for the GeoSPARQL compliance benchmark. Softw. Impacts. 8, 100071 (2021). https://doi.org/10.1016/j.simpa.2021.100071

Smals
ICT for society

Conclusions and Future Work

Conclusions

- While SHACL is powerful, one also has to consider sharing and reusing constraint components that may hold in many domains, applications,…

- We demonstrate how we can tackle this problem (method) for GeoSPARQL (the domain) with GeoSHACL. GeoSHACL shares SHACL Constraint Components according to Linked Data principles for GeoSPARQL.

- Our contribution seems simple, but we provide an important first step towards realizing this

Future work

- Incorporate functions beyond the simple features relation family

- Constraints (stored separately) to test an endpoint's GeoSPARQL capabilities

Smals
ICT for society

Smals Research

https://www.smalsresearch.be/

Christophe Debruyne

christophe.debruyne@smals.be