

# Comments from GeoSciML

[www.geosciml.org](http://www.geosciml.org)



# Background

- GeoSciML 2004-present
- GeoSciML v3 release imminent
- GML Application Schema for geological data
- Deployed using WFS by 10-20 geological surveys
- 4 testbeds
- Basis for INSPIRE GE theme

# Summary

- Nillable vs Voidable
- How to serialize nils
- Scoped names, controlled concepts
- Linking between features & observations
- Querying property values in services (WFS)

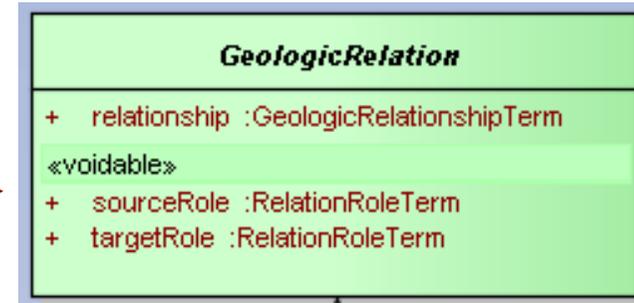
**Nillable/voidable**



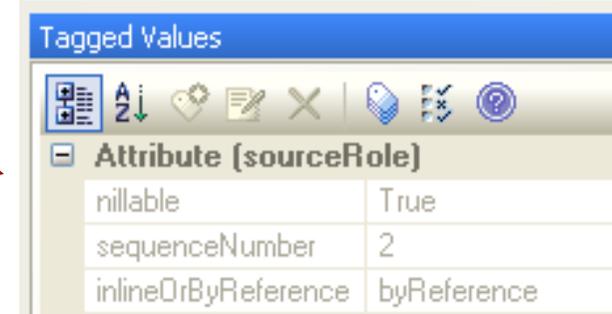
# “Voidable” stereotype vs “Nillable” tagged value

## In UML models:

INSPIRE uses the stereotype “voidable” to indicate that a property may carry a nil value (eg, `xsi:nil=“true” nilReason=“missing”`).



GeoSciML v2 follows GML 3.2 and uses the tagged value “nillable” to indicate that a property may carry a nil value.



The INSPIRE “voidable” stereotype is visible in UML diagrams, but only the “nillable” tagged value is automatically encoded as `<nillable=“true”>` in a GML schema by FullMoon.  
ie,

```
<element name="sourceRole" type="gml:ReferenceType" nillable="true">
```



## “Voidable” stereotype vs “Nillable” tagged value

GeoSciML v3 employs both the “voidable” stereotype and the “nillable” tagged value to achieve both UML visibility and automated GML schema encoding, but there is no robust link between the stereotype and the tagged value, leaving the potential for UML-to-GML errors.

There should be a single agreed way to indicate that a property is nillable in UML (ie, that is both visible in UML diagrams and is encoded in GML schemas) and how that should be processed by UML-to-GML tools.

# Serializing nils



# Serializing nil property values

There have been various ways considered for serializing nil/void property values (see <https://www.seegrid.csiro.au/wiki/AppSchemas/NilValues>)

- 1.Optional element
- 2.Element or attribute type="gml:nilReasonType"
- 3.Element with nillable="true" and @nilReason
- 4.Special value for xlink:href on complex-type GML property element
  - e.g. xlink:href="urn:ogc:def:nil:ogc:unknown"

GeoSciML previously used 4 but now adopting 3. What is best practice?

(A related more cosmetic issue is that the gml:AssociationAttributeGroup is often used where we want to enable a property to be by reference but this brings the gml:nilReason attribute in with it which isn't appropriate for non-nillable properties. Could define our own group but does gml:nilReason really belong to this convenience group?)

Referring to vocabulary items



# URI' s, codeSpace, and Referencing Controlled Vocabulary Concepts

During development of GeoSciML v2, there was considerable discussion about the appropriate use of the “codeSpace” attribute.

“CodeSpace” was used in some places to denote the method of resolving a URI (either a URN or HTTP-URI) delivered in an element content. eg,

```
<gml:name codeSpace="http://www.ietf.org/rfc/rfc2616">http://gsv-ws.dpi.vic.gov.au/resource/uri-cgi/feature/gsv/geologicunit/16777549126931109</gml:name>
```

```
<gml:name codeSpace="http://www.ietf.org/rfc/rfc2141">urn:cgi:feature:GSV:GeologicUnit:16777549126931109</gml:name>
```

At other times, “CodeSpace” was used to deliver an identifier for a controlled vocabulary. eg,

```
<gsml:rank codeSpace="http://www.dpi.vic.gov.au/earth-resources">Formation</gsml:rank>
```



# URI' s, codeSpace, and Referencing Controlled Vocabulary Concepts

In GeoSciML v3, we have agreed to:

- always use HTTP-URI for identifiers
- use “codeSpace” to identify controlled vocabularies
- N.B. It does not identify URI resolution methods

```
<swe:Category>  
  <swe:identifier>http://resource.ga.gov.au/classifier/ga/lithologygroup/0099</swe:identifier>  
  <swe:label>igneous felsic intrusive</swe:label>  
  <swe:codeSpace xlink:href="http://www.geoscience.gov.au/classifierscheme/ga/2012/lithologygroup"/>  
</swe:Category>
```

- use “xlink:href” and “xlink:title” as our GeoSciML community schema pattern for delivering controlled concepts

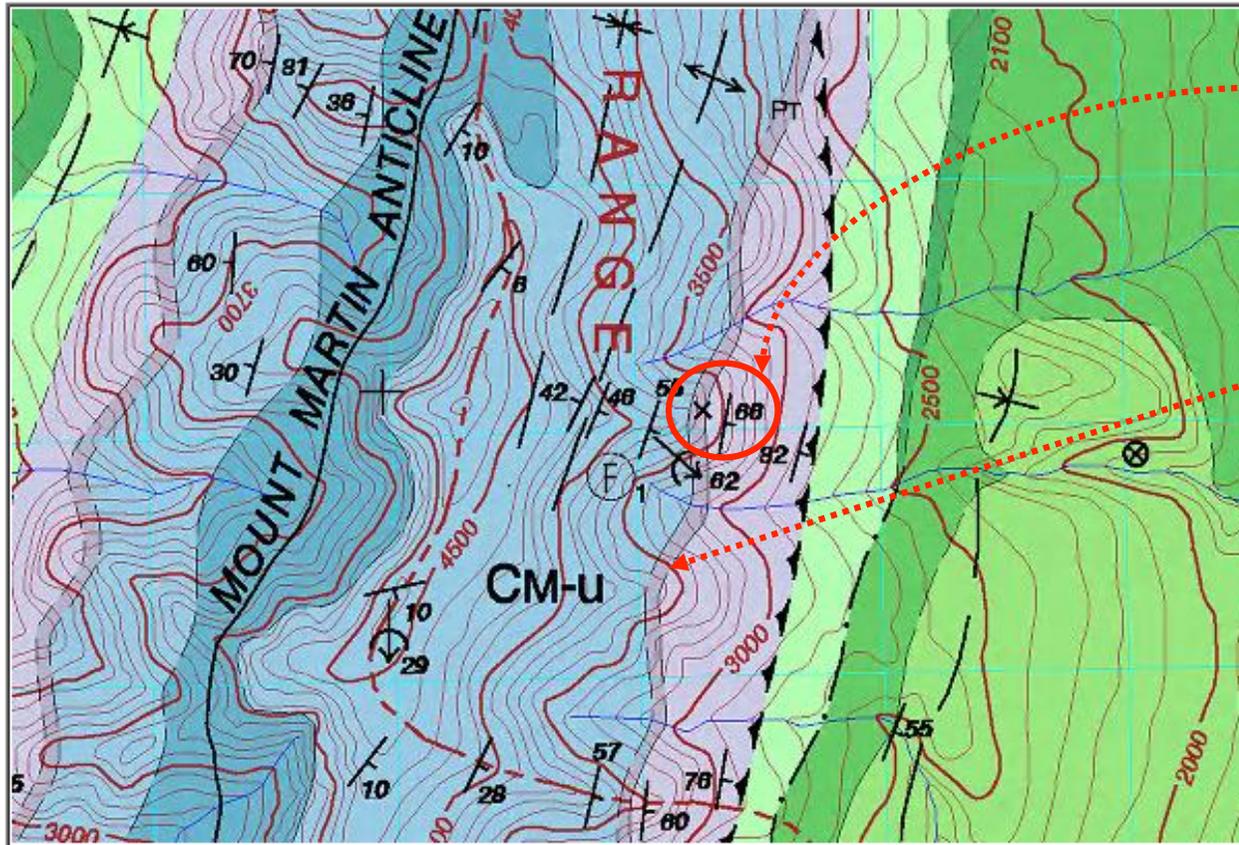
```
<gsmgu:geologicUnitType xlink:href="http://resource.geosciml.org/classifier/cgi/geologicunittype/0011/" xlink:title="Lithodemic Unit"/>
```

# GeoSciML & O&M



# GeoSciML is a feature view

- *GeoSciML is a feature view of geoscience*
- Geoscience relies on numerous observations to compile feature geometry and property values

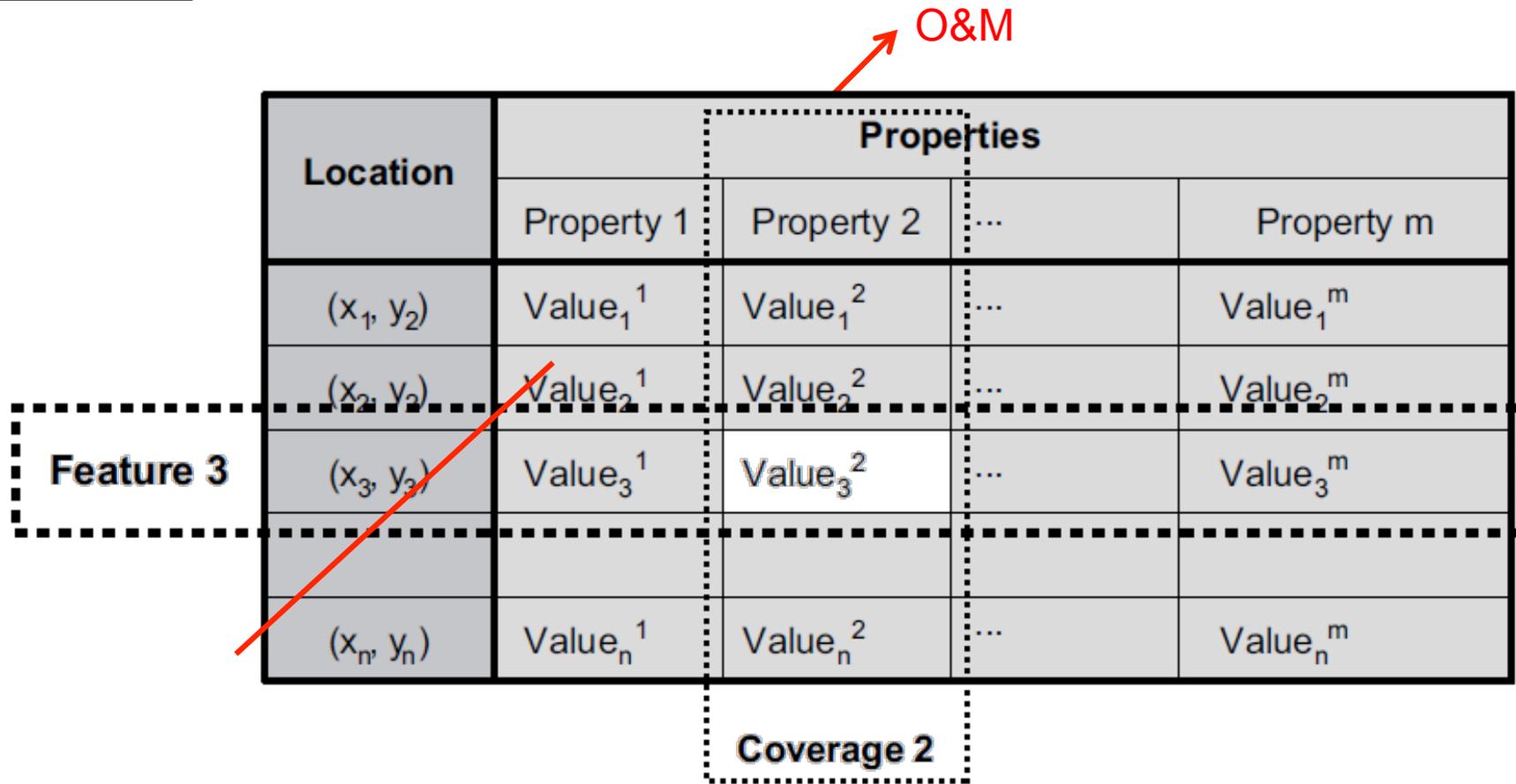


Observation site and Structural measurements

Sampled Feature (here, the contact)



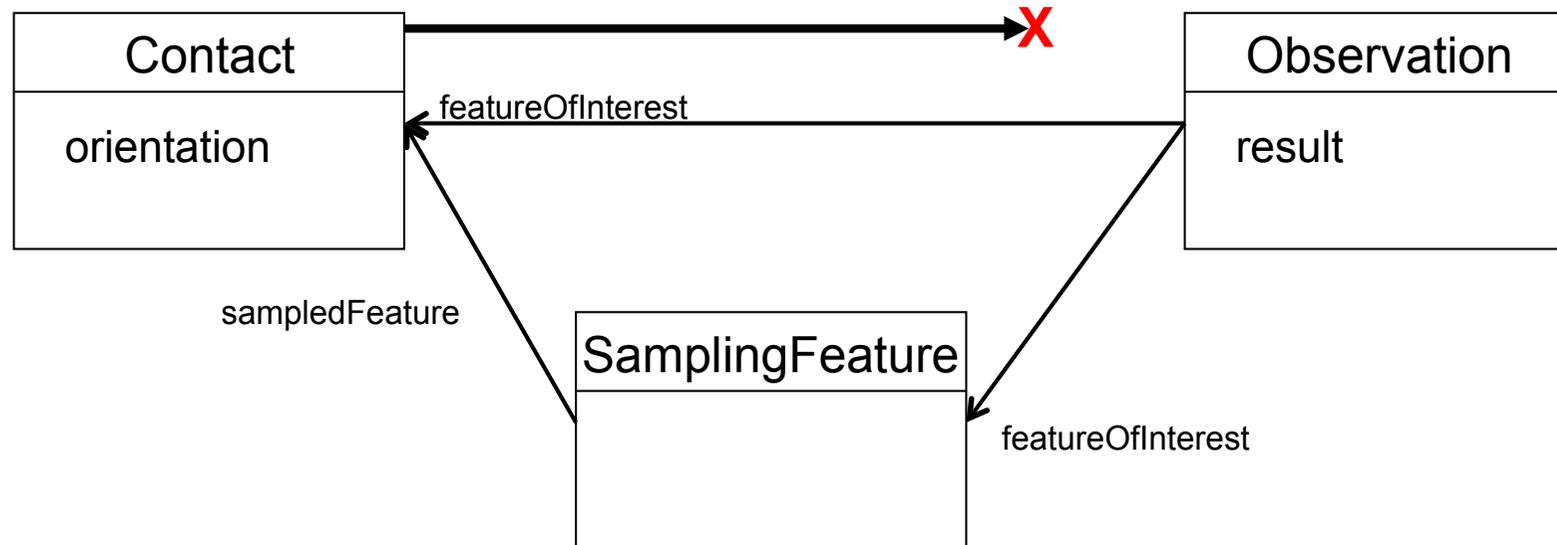
# O&M view point



- *The observation instance provides “metadata” for the property value-estimation process. (OGC 10-004r3)*



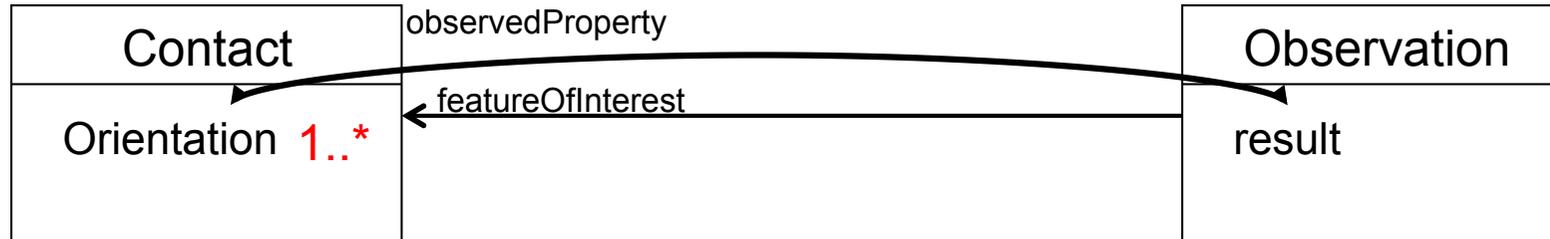
# Linking feature and observation views



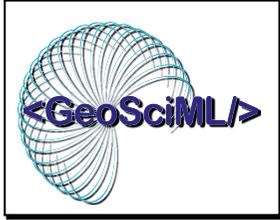
The Feature does not know about observations. The architecture suggests that the Observation repository is queried for a specific Feature (not the other way around)



# Gap in O&M linkage model



- Properties don't have identities !
  - you can't say this observation is about this property instance
  - This becomes obvious when the property can appear more than once (which observations is about which property value?)



# Property metadata

- Since property values don't have identity, we need to traverse from the property to the observation
- This would also resolve the Feature to Observation traversal

# Querying GeoSciML (WFS)



## Querying elements that may be inline or byReference

how to query feature properties that might be encoded in different ways:

- 1.inline "GML style" XML
- 2.xlink reference that dereferences to appropriate GML/XML element
- 3.xlink URI that identifies concept and might dereference to linked data style RDF



# Example

Schema: `<element name="samplingFrame" type="gml:ReferenceType"/>`

## •Instance:

- By reference: `<gsml:samplingFrame xlink:href="http://resource.geosciml.org/feature/bgs/EarthBedrockSurf" xlink:title="Bedrock surface"/>`

## In line

```
<gsml:samplingFrame>
  <SF_SamplingSurface>
    ... Other elements...
  ...
  </SF_SamplingSurface>
</gsml:samplingFrame>
```

## Options to filter:

- Xpath to `gsml:samplingFrame@xlink`
- Xpath to `gsml:samplingFrame/SF_SamplingSurface.../` some property of the the surface
  - Does the xpath refer to something inline
  - ...or does the filter processor follow xlink to get the referenced element?