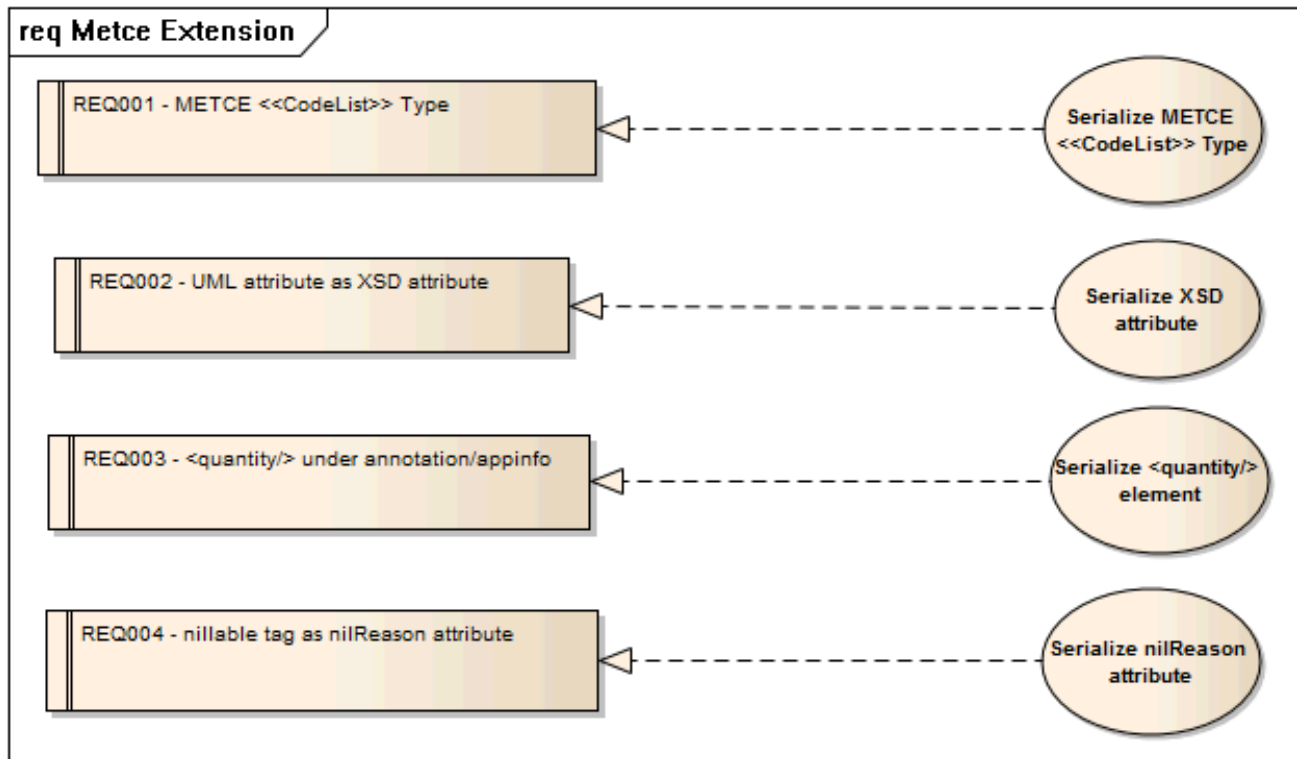


Feature Request on MDG:GML - Metce Extension Requirements

Diagram: Metce Extension



Use Case: Serialize METCE <<CodeList>> Type

Precondition: Stereotype and Tagged Value

- Class stereotype is <<CodeList>>.
 - Class tagged value "xsdEncodingRule" is "iso19136_2007_METCE_Extensions."
-

Scenarios:

Basic Path

Serialize METCE <<CodeList>> Type

1. The transformation on METCE <<CodeList>> class begins.

Alternate: 1a. Skip class serialization and log issue

2. The transformation checked the following tagged values are satisfied.

- "asDictionary" is true
- "vocabulary" and "extensibility" not empty

3. The transformation fetched the following values from the class.

- {Class.Name}; the class name
- {Class.TaggedValues.vocabulary.value}; the value of tagged value "vocabulary"
- {Class.TaggedValues.extensibility.value}; the value of tagged value "extensibility"
- {Class.Notes}; the notes from the class description
- {ApplicationSchema.xmlns} the namespace from the Application Schema

4. The transformation serialized the METCE <<CodeList>> Type to XSD as:

```
<element name="{Class.Name}" type="{ApplicationSchema.xmlns}:{Class.Name}Type"/>
<complexType name="{Class.Name}Type">
  <annotation>
    <appinfo>
      <vocabulary>{Class.TaggedValues.vocabulary.value}</vocabulary>
      <extensibility>{Class.TaggedValues.extensibility.value}</extensibility>
    </appinfo>
    <documentation>
      {Class.Notes}
    </documentation>
  </annotation>
  <complexContent>
```

```
<extension base="gml:ReferenceType"></extension>
</complexContent>
</complexType>
```

Alternate

Skip class serialization and log issue

1. The transformation checked the following tagged values are NOT satisfied.
 - a. "asDictionary" is true.
 - b. "vocabulary" and "extensibility" not empty.
2. The transformation skipped the serialization of the current class.
3. The transformation logged the following items to the output report.
 - a. The skipped class name

Use Case: Serialize XSD attribute

Precondition: Stereotype and Tagged Value

- Class stereotype are <<FeatureType>>, <<Type>> and <<DataType>>.
 - Class tagged value "xsdEncodingRule" is "iso19136_2007_METCE_Extensions."
-

Scenarios:

Basic Path

Serialize XSD attribute

1. The transformation on one of the following METCE class begins.
 - a. stereotype <<FeatureType>>
 - b. stereotype <<Type>>
 - c. stereotype <<DataType>>
2. The transformation processing an UML attribute and find the following tagged value existed.
 - a. "xsdAsAttribute" is true

Alternate: 2a. Fallback to XSD element and log issue
3. The transformation checked the UML attribute type belongs to one of the following.
 - a. <<Enumeration>> class
 - b. CharacterString
 - c. Boolean
 - d. Real
 - e. Number
 - f. Decimal
 - g. Date
 - h. Time
 - i. DateTime
 - j. Integer
4. The transformation fetched the following values from the attribute.
 - a. {Attribute.Name}: the attribute name
 - b. {Attribute.Type}: the attribute type
 - c. {Attribute.Type.Namespace}: the namespace of the attribute type
5. The transformation serialized the UML attribute to XSD attribute as:

<attribute name="{Attribute.Name}" type="{Attribute.Type.Namespace}:{Attribute.Type}"/>

Alternate

Fallback to XSD element and log issue

1. The transformation checked the UML attribute type NOT belongs to one of the following.
 - a. <<Enumeration>> class
 - b. CharacterString
 - c. Boolean
 - d. Real
 - e. Number
 - f. Decimal
 - g. Date
 - h. Time
 - i. DateTime
 - j. Integer

2. The tranformation fallback the UML attribute to XSD element as GML default.

3. The transformation logged the following items to the output report.
 - a. The processed class name
 - b. The fallback processed attribute name

Use Case: Serialize <quantity/> element

Precondition: Stereotype and Tagged Value

Class stereotype are <<Type>> and <<DataType>>.

Class tagged value "xsdEncodingRule" is "iso19136_2007_METCE_Extensions".

Scenarios:

Basic Path

Serialize <quantity/> element

1. The transformation on one of the following METCE class begins.
 - a. stereotype <<Type>>
 - b. stereotype <<DataType>>

2. The transformation processing an UML attribute and find the following tagged value exists.
 - a. "quantity" not empty

3. The transformation fetched the following values from the attribute.
 - a. {Attribute.Name}: the attribute name
 - b. {Attribute.Type}: the attribute type
 - c. {Attribute.Type.Namespace}: the namespace of the attribute type
 - d. {Attribute.TaggedValues.quantity.value}: the value of tagged value "quantity"
 - e. {Attribute.Notes}: the notes from the attribute description

4. The transformation serialized the UML attribute to XSD element as:

```
<element name="{Attribute.Name}" type="{Attribute.Type.Namespace}:{Attribute.Type}">
  <annotation>
    <appinfo>
      <quantity>{Attribute.TaggedValues.quantity.value}</quantity>
    </appinfo>
    <documentation>{Attribute.Notes}</documentation>
  </annotation>
</element>
```

Use Case: Serialize nilReason attribute

Precondition: Stereotype and Tagged Value

- Class stereotype is <<Type>>.
 - Class tagged value "xsdEncodingRule" is "iso19136_2007_METCE_Extensions."
-

Scenarios:

Basic Path

Serialize nilReason attribute

1. The transformation on METCE <<Type>> class begins.
2. The transformation checked the following tagged value exists.
 - a. "nillable" is true
3. The transformation add "nilReason" attribute to the complexType as:
`<attribute name="nilReason" type="gml:NilReasonType"/>`