All Fields marked with * are mandatory.

Change Request #:	**153**
Assigned OGC Document #:	**11-076**
Name:	*Jeroen Dries
Organization:	*Luciad NV
Email:	*jeroen.dries@luciad.com
Document Name/Version:	*Symbology Encoding Implementation Specification / 1.1.0
OGC Project Document:	*05-077r4

If this is a revision of a previous submission and you have a Change Request Number, then check here: □
Enter the CR number here:
Enter the Revsion Number that you are revising here:

Title:	*Symbolizer and Rule for styling of nested child objects
Source:	*OWS-8 Aviation
Work item code:	
Category:	* B (Addition of feature)

Reason for change: *

```
Complex data models like AIXM 5 have feature types that are a
composition of nested objects. The current SE 1.1 only allows limited
styling of these child objects, such as using the same style for all
objects. This change request defines a new type of rule and symbolizer
that makes this possible.
```

Summary of change: *

```
Add a new symbolizer: \'CompositeChildSymbolizer\' and a new type of
rule \'CompositeChildRule\'. Both have a mandatory element that points
to a list of nested child objects, and a list of symbolizers or rules
that will be applied to each object in the list. For instance, for a
GML feature that looks like:
<Parent>
   <ParentColor>Black</ParentColor>
   <Child>
      <Color>Blue</Color>
      <ChildGeometry> ...</ChildGeometry>
  </Child>
   <Child>
      <Color>Red</Color>
      <ChildGeometry> ...</ChildGeometry>
      <ChildGeometry> ...</ChildGeometry>
  </Child>
   <Child>
      <ChildGeometry> ...</ChildGeometry>
  </Child>
<Parent>
```

A composite child symbolizer would look like this:

```
<se:CompositeChildSymbolizer>
   <se:childProperty>Child</se:childProperty>
      <se:PolygonSymbolizer>

<se:Geometry><ogc:PropertyName>ChildGeometry</ogc:PropertyName></se:Geometry>
         <se:Stroke>
           <se:SvgParameter
name=\"stroke\"><ogc:PropertyName>Color</ogc:PropertyName></se:SvgParameter>

           <se:SvgParameter
name=\"stroke-width\">4</se:SvgParameter>
         </se:Stroke>
  </se:PolygonSymbolizer>
</se:CompositeChildSymbolizer>
```

Using the CompositeChildRule? we could in the same way define a rule that will only use the color from the child if it is not null. Note that the semantics for a composite child rule would be that each rule in the composite child rule would be treated as if it were a top level rule.

Applications that support this symbolizer will also need to support the XPath parent axes, or \'../\' in short to reference the parent of a child node from within the composite child rule or symbolizer.

The XML schema for these types would look like:

```
<xsd:element name=\"CompositeChildSymbolizer\"
type=\"se:CompositeChildSymbolizerType\"

    substitutionGroup=\"se:Symbolizer\">
    <xsd:annotation>
      <xsd:documentation>A
\"CompositeChildSymbolizer\" comprises a group of
Symbolizers that are to be applied to a list of
objects</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType
name=\"CompositeChildSymbolizerType\">
    <xsd:complexContent>
      <xsd:extension base=\"se:SymbolizerType\">
        <xsd:sequence>
          <xsd:element ref=\"fes:ValueReference\"/>
          <xsd:element ref=\"se:Symbolizer\"
maxOccurs=\"unbounded\"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

<xsd:element name=\"CompositeChildRule\"
type=\"se:CompositeChildRuleType\"
    substitutionGroup=\"se:Rule\">
    <xsd:annotation>
      <xsd:documentation>A \"CompositeChildRule\"
comprises a group of Rules that are to be applied to a list of
objects</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name=\"CompositeChildRuleType\">
    <xsd:complexContent>
      <xsd:extension base=\"se:RuleType\">
        <xsd:sequence>
          <xsd:element ref=\"fes:ValueReference\"/>
          <xsd:element ref=\"se:Rule\"
maxOccurs=\"unbounded\"/>
        </xsd:sequence>
      </xsd:extension>
```

	`</xsd:extension>`
	`</xsd:complexContent>`
	`</xsd:complexType>`

Consequences if not approved:	Styling of complex objects will not be possible, the example given is derived from AIXM 5, where we want to use this type of rule/symbolizer in at least 30 feature types out of about 100 features types with geometry in total. As there is no alternative to the proposed feature, custom extensions would be needed.
Clauses affected:	* 10 Rules 11 Symbolizers
Additional Documents affected:	
Supporting Documentation:	
Comments:	
Status:	Assigned
Assigned To:	SLDSE 1.2 SWG
Disposition:	Referred