**Title:** Symbolizer and Rule for styling of nested child objects

**Source:** OWS-8 Aviation

**Category:** 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>
  </Child>
</Parent>
```
A composite child symbolizer would look like this:

```xml
<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:

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

```xml
<xsd:element name="CompositeChildRule" type="se:CompositeChildRuleType">
  <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: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:complexContent>
  </xsd:complexType>
</xsd:element>
```
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.

<table>
<thead>
<tr>
<th>Clauses affected:</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 Rules</td>
</tr>
<tr>
<td></td>
<td>11 Symbolizers</td>
</tr>
</tbody>
</table>

Additional Documents affected:

Supporting Documentation:

Comments:

Status: Assigned

Assigned To: SLDSE 1.2 SWG

Disposition: Referred