Report: Some Unresolved Issues with the OGC Symbology Encoding (SE)

UTP1146-01 | 18 July 2008 | Status: Draft

Submitted to OGC

Tenet Technology Ltd

North Heath Lane, Horsham, West Sussex, RH12 5UX, United Kingdom

www.tenettechnology.com

email: neil.kirk@tenettechnology.com

*Tel: +44 01403 273173 * Fax: +44 01403 273123*

Registered in England No. 2348404

Section 1 Introduction

1.1 Overview

TENET Technology Ltd (TENET) understand that The United Kingdom Hydrographic Office (UKHO) have a requirement to specify symbology for AML [9] in an open standard which will allow easier and more consistent adoption of the default symbology set by implementers of AML. The Open Geospatial Consortium (OGC) Symbology Encoding (SE) is an XML-based language for defining visual portrayal rules of geographic data. This language is an open standard intended to be used, amongst other things, to portray the output of OGC Web Services. As such it is an ideal candidate for use with modern GIS systems based on open system architecture principles.

A number of symbology standards for geographic data are already available, in particular, the National Geospatial-Intelligence Agency (NGA) specification MIL-PRF-89045: *Geospatial Symbols for Digital Displays (GeoSym)* [3]; the International Hydrographic Organization (IHO) special publication S-52: *Specifications for Chart Content and Display Aspects of ECDIS* [8] [1]; and the Department of Defense (DoD) standard MIL-STD-2525B: *Common Warfighting Symbology* [4].

The UKHO have compiled a draft symbology for AML by attempting for find a suitable match for the AML entities in each of the symbology standards GeoSym, S-52 and MIL-STD-2525B. The majority of symbol assignments make use of S-52 and, to a lesser extent, GeoSym. This is not surprising given the nature of AML and S-52.

1.2 Purpose

TENET has undertaken experimental work to implement an OGC Symbology Encoding of the UKHO draft AML Symbology Guidance specification [1]. As part of this work several issues and restrictions of the standards used were identified.

This document summarises some of the issues relating to the OGC SE Implementation Specification [6]. It is essentially an extract from [10] that has been compiled and made available to the OGC community for wider discussion. Section 2 is background material relating to the OGC SE and the UKHO AML symbology. Section 3 lists the issues which are centred on the encoding of S-52 symbols using OGC SE as this is the standard predominantly used by the AML symbology. However, it is expected that similar issues arise with other symbology standards such as GeoSym.

---

[1] Of particular relevance are the *S-52 Colours & Symbols Specifications* (S-52 Appendix 2) and the *S-52 Presentation Library* (Annex A to S-52 Appendix). However, it should also be noted that the S-52 standard is not just a symbol specification, but a complete and extensive specification for configuring the display aspects of an ECDIS and ENC data.

1.3 Abbreviations

AML	Additional Military Layers
DoD	Department of Defense
ECDIS	Electronic Chart Display and Information System
ENC	Electronic Navigational Chart
GIS	Geographic Information System
GML	Geography Markup Language
IHO	International Hydrographic Organization
NGA	National Geospatial-Intelligence Agency
OGC	Open Geospatial Consortium
SE	Symbology Encoding
SLD	Styled Layer Descriptor
SVG	Scalable Vector Graphics
UKHO	United Kingdom Hydrographic Office
VMAP	Vector Map
VPF	Vector Product Format
WMS	Web Map Service
XML	eXtensible Markup Language

1.4 References[2]

1. AML Symbology Guidance, Version 1.0, UKHO.

2. ISO/TC 211 N2167, Revision of ISO 19117:2005: Geographic information – Portrayal, 2007.

[2] Versions of the symbology standards GeoSym, S-52 and MIL-STD-2525B are those referenced by the UKHO AML Symbology Guidance, Version 1.0 [1]. In some cases later versions of these standards have been published.

3. MIL-DTL-89045, Geospatial Symbols for Digital Displays (GeoSym), NGA, 2004.

4. MIL-STD-2525B, Common Warfighting Symbology, DoD, 1999.

5. OGC 04-095, Filter Encoding Implementation Specification, v.1.1.0, 2007.

6. OGC 05-077r4, Symbology Encoding Implementation Specification, v.1.1.0, 2006.

7. OGC 05-078r4, Styled Layer Descriptor profile of the Web Map Service Implementation Specification, v.1.1.0, 2007.

8. S-52, Specifications for Chart Content and Display Aspects of ECDIS, IHO, 5th Edition, 1996 (amended 1999).

9. STANAG 7170, Additional Military Layers.

10. UTP1140-01, Specifying AML Symbology using the OGC Styled Layer Descriptor (SLD) / Symbology Encoding (SE), TENET, 2008.

1.5 Document History

Issue	Date	Author	Comments
01	18 July 2008	N. P. Kirk	First draft

Section 2 A Summary of the OGC SE

The Symbology Encoding (SE) language is defined in a companion specification [6] to the OGC Styled Layer Descriptor (SLD) specification [7]. The latter is actually a profile of the OGC Web Map Service (WMS) specification, defining new operations for the OGC services in addition to the styling language. A "Styled Layer Descriptor" provides a description of the map styling available for the various layers served by a WMS. In the transition from version 1.0.0 to 1.1.0, the SE definition was decoupled from the SLD specification in recognition of the fact that it may be of use in settings other than OGC services. Although the term "SLD" is often used to refer to such styling, it is the SE language that is used to specify the styling and portrayal rules.

OGC SE provides an XML grammar for specifying portrayal symbols and rules, together with the semantics necessary for a GIS to interpret the encoding and render the geospatial information. This section provides a summary of the OGC SE, in particular those constructs that are used to encode the AML Symbology. For more details the reader is referred to [6].

The FeatureTypeStyle element defines the styling that is to be applied to a single feature type. Its XML Schema definition is as follows:

```
<xsd:element name="FeatureTypeStyle" type="se:FeatureTypeStyleType"/>

<xsd:complexType name="FeatureTypeStyleType">
  <xsd:sequence>
    <xsd:element ref="se:Name" minOccurs="0"/>
    <xsd:element ref="se:Description" minOccurs="0"/>
    <xsd:element ref="se:FeatureTypeName" minOccurs="0"/>
    <xsd:element ref="se:SemanticTypeIdentifier" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="se:Rule"/>
      <xsd:element ref="se:OnlineResource"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="version" type="se:VersionType"/>
</xsd:complexType>
```

Figure 1: The SE FeatureTypeStyle Element

The Name and Description elements are used to identify the feature style. The Name element is typically a label or token that provides a unique identifier, whereas the Description element is for human-readable information. The FeatureTypeName identifies the feature type (as defined in an accompanying feature catalogue or application schema) that the style is for. The SemanticTypeIdentifier is experimental – it is used with the AML Symbology to address the issue of delineation; this is discussed in Section 3.3.

An example instance for the point feature type BCNCAR (Beacon, cardinal) is shown in Figure 2.

```xml
<se:FeatureTypeStyle>
  <se:Name>BCNCAR-P</se:Name>
  <se:Description>
    <se:Title>AML feature portrayal for: Beacon, cardinal</se:Title>
  </se:Description>
  <se:FeatureTypeName>BCNCAR</se:FeatureTypeName>
  <se:SemanticTypeIdentifier>delineation:point</se:SemanticTypeIdentifier>
  <!-- Multiple se:Rule elements corresponding to the four portrayal rules … -->
</se:FeatureTypeStyle>
```

Figure 2: FeatureTypeStyle for BCNCAR

The main body of a FeatureTypeElement is a sequence of portrayal rules; these rules are applied to a feature instance to determine what symbol to use. In an SE instance document, portrayal rules are encoded using Rule elements, whose XML Schema definition is shown below.

```xml
<xsd:element name="Rule" type="se:RuleType"/>

<xsd:complexType name="RuleType">
  <xsd:sequence>
    <xsd:element ref="se:Name" minOccurs="0"/>
    <xsd:element ref="se:Description" minOccurs="0"/>
    <xsd:element ref="se:LegendGraphic" minOccurs="0"/>
    <xsd:choice minOccurs="0">
      <xsd:element ref="ogc:Filter"/>
      <xsd:element ref="se:ElseFilter"/>
    </xsd:choice>
    <xsd:element ref="se:MinScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:MaxScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:Symbolizer" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 3: The SE Rule Element

The Name and Description elements are used, as before, for identification purposes. The Filter element is used to encode a rule expression: a sequence of attribute conditions that are evaluated to see if the associated symbol should be used to portray the feature instance. Alternatively, the ElseFilter element may be used to indicate a catch-all "else clause" for features not matching any of the given rule expressions. The Symbolizer element specifies how to draw the symbol. Various sub-types of Symbolizer are available (Line, Polygon, Point, Text, Raster) for encoding point graphics, simple linestyles, complex linestyles, coloured fills, patterned fills, text labels, etc. The Filter specification is described in more detail in [5] and the Symbolizer in Section 11 of the OGC SE specification [6].

The Rule element may also provide other information such as the minimum and maximum map-rendering scales for which the rule should be applied and an explicit graphic to be displayed in a legend for the rule, for example by a WMS. These features are not presently used in the AML Symbology.

Figure 4 shows an example rule for the feature type BCNCAR. This can be summarised as: “if category of cardinal mark is east (CATCAM == 2) then render the point symbol specified by the external bitmap graphic file BCNCAR02.bmp at the location indicated by the feature property *geometry*”.

```
<se:Rule>
 <se:Description>
  <se:Title>Portrayal rule corresponding to AML (real-world) entity: Beacon, cardinal (East)</se:Title>
 </se:Description>
 <ogc:Filter>
  <ogc:PropertyIsEqualTo>
   <ogc:PropertyName>CATCAM</ogc:PropertyName>
   <ogc:Literal>2</ogc:Literal>
  </ogc:PropertyIsEqualTo>
 </ogc:Filter>
 <se:PointSymbolizer>
  <se:Geometry>
   <ogc:PropertyName>geometry</ogc:PropertyName>
  </se:Geometry>
  <se:Graphic>
   <se:ExternalGraphic>
    <se:OnlineResource xlink:type="simple" xlink:href="GraphicFiles/S-52/BCNCAR02.bmp"/>
    <se:Format>image/bmp</se:Format>
   </se:ExternalGraphic>
  </se:Graphic>
 </se:PointSymbolizer>
</se:Rule>
```

Figure 4: Rule for BCNCAR

Symbolizers are part of the Rule and either reference out to an external graphics file, as shown for the point symbolizer in Figure 4, or use SVG commands, as shown for the simple linestyle symbolizer for the area delineation of ADMARE in Figure 5. Composite symbols, such as “SY(LIGHTDEF) + SY(PILPNT02)” for feature type “LIGHTS” are represented by multiple symbolizers attached to the Rule.

```
<se:LineSymbolizer>
 <se:Geometry>
  <ogc:PropertyName>geometry</ogc:PropertyName>
 </se:Geometry>
 <se:Stroke>
  <se:SvgParameter name="stroke">#9EA198</se:SvgParameter>
  <se:SvgParameter name="stroke-width">2</se:SvgParameter>
  <se:SvgParameter name="stroke-dasharray">12.0 6.0</se:SvgParameter>
 </se:Stroke>
</se:LineSymbolizer>
```

Figure 5: Line Symbolizer LS(DASH,2,CHGRF) for ADMARE (area)

The source for these symbology rules is the UKHO draft AML Symbology Guidance [1]. A subset of these rules, collated as part of the TENET study, is shown in Table 1.

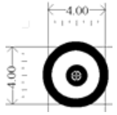
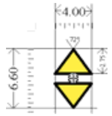
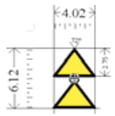
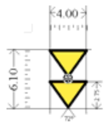
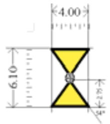
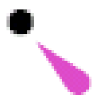
AML (REAL-WORLD) ENTITY	AML FEATURE CLASS	AML SYMBOL RECOMMENDATION	SYMBOLIZER SOURCE SPECIFICATION	SYMBOLIZER SOURCE ID	FEATURE TYPE NAME	DELINEATION	RULE FILTER
port	Administration area (named)		S-52	SY(POSGEN03)	ADMARE	P	catadm=501
territorial land area	Administration area (named)		S-52	LS(DASH,2, CHGRF)	ADMARE	A	catadm=502
Beacon, cardinal (East)	Beacon, cardinal		S-52	SY(BCNCAR02)	BCNCAR	P	CATCAM=2
Beacon, cardinal (North)	Beacon, cardinal		S-52	SY(BCNCAR01)	BCNCAR	P	CATCAM=1
Beacon, cardinal (South)	Beacon, cardinal		S-52	SY(BCNCAR03)	BCNCAR	P	CATCAM=3
Beacon, cardinal (West)	Beacon, cardinal		S-52	SY(BCNCAR04)	BCNCAR	P	CATCAM=4
Light	Light		S-52	SY(LIGHTDEF) + SY(PILPNT02)	LIGHTS	P	<none>

Table 1: TENET Version of AML SYMBOLOGY_MATRIX

A FeatureTypeStyle element may provide multiple Rule elements, having overlapping rule expressions. Similarly a Rule element may contain multiple Symbolizers. In such cases the ordering is important and follows the "painters model" with the first item in a list being the first item rendered and hence being on the "bottom" of the display. This mechanism allows complex symbology to be constructed, both in terms of multiple, overlapping Rules, and composite Symbolizers.

Section 3 Unresolved Issues with the OGC SE

3.1 Introduction

This section summarises some of the issues encountered in attempting to encode the UKHO's draft symbology for AML using OGC SE. The issues raised are centred on the encoding of S-52 symbols as this is the standard predominantly used by the AML symbology. This section only summarises the issues found – it is not meant to be a comprehensive analysis of encoding the whole of S-52 using OGC SE.

3.2 Interpretation of External Graphic Symbols

While SE makes use of well-known styling parameters based on SVG, these only really cover the simple linestyles and fillstyles. In order to represent point symbols (other than basic named shapes), complex linestyles and patterned fillstyles SE provides elements for referencing graphics defined in external files. Several questions have been raised whilst trying to implement S-52 constructs via external graphic files. Some specific issues are presented below.

Probably the most troublesome area is that of complex linestyles. The SE GraphicStroke element is used to specify a repeated-linear-graphic linestyle. It references an external linear graphic requiring two "hot-spot" points within the space of the graphic to identify the start and end of the line segment. Its structure is shown in Figure 6 and the structure of the Graphic element is shown in Figure 8; for further details refer to Section 11 in [6].

```
<xsd:element name="GraphicStroke" type="se:GraphicStrokeType"/>

<xsd:complexType name="GraphicStrokeType">
  <xsd:sequence>
    <xsd:element ref="se:Graphic"/>
    <xsd:element ref="se:InitialGap" minOccurs="0"/>
    <xsd:element ref="se:Gap" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 6: The SE GraphicStroke Element

In S-52 a complex linestyle is formed from a repeating graphic symbol whose definition is similar to the symbol definition for a point. The linestyle symbol has its own pivot point about which it is rotated so that it is oriented in the direction of the line segment it represents. S-52 uses two types of complex linestyle symbol:

1. *Single unit type*: this linestyle consists of a single repeating graphic symbol which is concatenated to form a string of symbols between two vertices of the line, using one orientation. Implemented verbatim, this type of complex linestyle can only symbolise a straight line. In order to change

orientation at a vertex an additional *simple* linestyle is required to fill any gap between the last symbol and the vertex; a dashed style is typically used.

2. *Composite type*: this linestyle uses a composite graphic symbol, constructed from a sequence of *multiple* sub-symbols and horizontal lines. This style is more suitable for rendering non-linear curves: the composite graphic symbol being repeated along the line but, being constructed from smaller symbols and horizontal lines, can change its orientation at the line's vertices. This complex linestyle can be thought of as a simple linestyle with additional symbols rendered at defined points along the line.

   For example, the S-52 linestyle for the boundary of an anchorage area, Figure 7, consists of a horizontal line, a 'V' symbol, an anchor symbol, followed by two more horizontal line and 'V' symbol pairings. Note that the only vectors in this linestyle are the four horizontal dashes, the other components are symbols. Also note that a "pen up" instruction is required to change the drawing position before drawing each successive symbol – the next symbol does not start at the inferred end position of the previous symbol/line.

Figure 7: S-57 Linestyle LC(ACHARE51)

The general problem is not necessarily with the SE specification. It is possible that by using a vector graphics file format such as SVG that the above problems could be solved using the functionality provided by that format, i.e., replicating the drawing commands to render the linestyle. Further analysis is required to ascertain whether such linestyles can be implemented effectively and efficiently by an appropriate choice of external graphic format. It is clear that a general solution, that does not target a specialist vector format, is not achievable without extensions to the SE GraphicStrokeType. While it is not the responsibility of SE to mandate the use of a particular format, the specification would benefit from several implementation annexes, or even profiles, that demonstrate the use of a few common, widely supported graphics file formats. This would identify any shortcomings in the SE specification and also establish best practices for addressing the above problems. Without this, it is envisaged that system implementers would be burdened with supporting numerous "solutions" and formats and interoperability would suffer.

A general observation regarding the SE Graphic element is the requirement for a "hot spot", rotation point, etc., in the external graphic, either inherently in the external format or via the defaults defined in the SE specification (the SE default is to use a system-dependent "central point" for placement, the centroid for rotation, etc.). Similarly, every S-52 symbol has its own *pivot point* (the S-52 terminology) that is used to position the symbol and around which the symbol is scaled and rotated. This is therefore directly relevant to the implementation of S-52 symbols.

Rather than rely on the external graphic format to provide these points, they could be defined explicitly in the SE encoding. This approach is more transparent and flexible and has some precedent as the SE Graphic element, Figure 8, has an AnchorPoint, Size, Rotation and Displacement that allow more control over the placement of the symbol. However, the SE specification is unclear about the use of AnchorPoint and its equivalence with the hot spot. In addition, the AnchorPoint is restricted to the interior of the symbol bounding box and is not used when rotating the symbol. These issues need to be resolved to define S-52 pivot points effectively. For example, an S-52 symbol with a pivot point *outside* of its bounding box cannot be encoded using an external raster bitmap without the ability to specify the pivot point within the SE itself.

```
<xsd:element name="Graphic" type="se:GraphicType"/>

<xsd:complexType name="GraphicType">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="se:ExternalGraphic"/>
      <xsd:element ref="se:Mark"/>
    </xsd:choice>
    <xsd:element ref="se:Opacity" minOccurs="0"/>
    <xsd:element ref="se:Size" minOccurs="0"/>
    <xsd:element ref="se:Rotation" minOccurs="0"/>
    <xsd:element ref="se:AnchorPoint" minOccurs="0"/>
    <xsd:element ref="se:Displacement" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 8: The SE Graphic Element

3.3 Portraying a Feature Based on its Delineation

The feature portrayal rules in S-52 and GeoSym are dependent on the dimensional classification of the feature (point, line, area). ISO 19117 uses the term *delineation* to refer to this concept and defines it as *the geometric dimension associated with the spatial attribute that is used to portray a feature*. It is clear from all these standards that feature portrayal is delineation specific, not mixing, for example, the portrayal of points and lines.

The following S-52 examples are used by the AML Symbology and demonstrate the issue. According to the AML Feature Catalogue, the feature type Marine Farm/Culture (MARCUL) is allowed the following delineations.

Area delineation: portrayed using patterned fill AP(MARCUL02) and complex linestyle LC(NAVARE51).

Figure 9: Portrayal of Marine Farm/Culture (MARCUL) – Area Delineation

Line delineation: portrayed using a dashed, grey simple linestyle LS(DASH,2,CHGRF).

Figure 10: Portrayal of Marine Farm/Culture (MARCUL) – Line Delineation

Point delineation: portrayed using the symbol SY(MARCUL02).

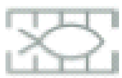


Figure 11: Portrayal of Marine Farm/Culture (MARCUL) – Point Delineation

However, the OGC SE FeatureTypeStyle element (Figure 1) is missing a mechanism to specify the delineation. It is therefore not possible to define feature portrayals that differentiate between different delineations, as with the cases illustrated above.

This problem does not always manifest itself. With VPF products such as VMAP, individual feature types are defined according to each delineation as the feature-attribute binding may vary. For example, separate feature types (PAL015, LAL015, AAL015, say) are required for the point, line and area representations of the feature concept AL015 – Building. However, for ENC and AML a single feature type is defined which may support one or more delineations.

The following fixes/workarounds have been considered. Solution 1 is the most obvious but Solution 3 is based on the present SE specifications and schemas so has been adopted for the AML SE encoding.

1. Add a 'Delineation' element to FeatureTypeStyleType in line with ISO 19117:

```
<se:FeatureTypeStyle>
  <se:Description>
    <se:Title>AML feature portrayal for: Administration area (named)</se:Title>
  </se:Description>
  <se:FeatureTypeName>ADMARE</se:FeatureTypeName>
  <se:Delineation>area</se:Delineation>
  <!-- Rules -->
</se:FeatureTypeStyle>
```

   However, this requires a change to the OGC XML Schemas or a bespoke extension that no longer resides in the OGC namespace. This is possibly a future solution rather than present.

2. As a workaround that uses the current OGC SE XML Schemas the FeatureTypeName element could be extended to include the delineation as structured text:

```
<se:FeatureTypeStyle>
  <se:Description>
    <se:Title>AML feature portrayal for: Administration area (named)</se:Title>
  </se:Description>
```

```
  <se:FeatureTypeName>ADMARE:area</se:FeatureTypeName>
  <!-- Rules -->
</se:FeatureTypeStyle>
```

Although there is some precedent for this approach, it does represent a specialist encoding, that "extends" the semantics of se:FeatureTypeName.

3. This solution is similar to 2 except that it uses the experimental se:SemanticTypeIdentifier element to encode the delineation.

```
<se:FeatureTypeStyle>
  <se:Description>
    <se:Title>AML feature portrayal for: Administration area (named)</se:Title>
  </se:Description>
  <se:FeatureTypeName>ADMARE</se:FeatureTypeName>
  <se:SemanticTypeIdentifier>delineation:area</se:SemanticTypeIdentifier>
  <!-- Rules -->
</se:FeatureTypeStyle>
```

This is a good candidate solution as it means the encoding is compliant with the existing OGC specification, and the "extension" to capture the delineation has been encoded using what OGC describe as an "experimental" element – there is some scope for interpretation based on use. A client that knows the "extension" semantics can handle the SE correctly, a client that doesn't would only get it "slightly" wrong (if multiple FeatureTypeStyles are provided for a given feature type, for multiple delineations, then this client would render both styles rather than a single style).

4. Another possible solution would be to add the delineation condition to the Rule's Filter clause. The problem is that OGC Filter provides operations for comparing property values, not type. We really want to write something like 'geometry.isKindOf(GM_Surface)' where 'geometry' is a property of the feature. Arguably this is UML/OCL and the condition should use XPath and GML constructs to be part of OGC Filter. A named function, that serves the purpose of the OCL method 'isKindOf', is probably required as part of the OGC SE specification.

This solution is rather complicated to specify (and implement) so a wider discussion should be sought before pursuing it.

5. A slightly different take on 4 that uses existing OGC Filter elements is the following:

```
<se:FeatureTypeStyle>
  <se:Description>
    <se:Title>AML feature portrayal for: Administration area (named)</se:Title>
  </se:Description>
  <se:FeatureTypeName>ADMARE</se:FeatureTypeName>
  <se:Rule>
    <ogc:Filter>
      <ogc:Not>
        <ogc:PropertyIsNull>
          <ogc:PropertyName>aml:geometry/gml:Polygon</ogc:PropertyName>
        </ogc:PropertyIsNull>
      </ogc:Not>
```

```xml
    </ogc:Filter>
    <!-- Symbolizers -->
  </se:Rule>
  <!-- Rules -->
</se:FeatureTypeStyle>
```

However, within the context of the OGC Filter semantics this is questionable: 'aml:geometry/gml:Polygon' is the property **value**, 'aml:geometry' is the **property**. Also, from an XPath viewpoint is "aml:geometry/gml:Polygon" NULL when aml:geometry has a different value such as "aml:geometry/gml:Point"? It is therefore questionable whether this kind of expression is supported by standard implementations of OGC Filter. In addition, it does not fully express the condition "is delineation equal to area" as it specifies concrete GML types such as gml:Polygon.

As an aside: this raises the question of how to determine the dimensional classification of a feature. For many existing geo-spatial products such as VMAP, AML, ENC a feature instance is encoded as a geometrical object so the dimension is explicit. However, in the ISO 19100 setting, the paradigm is changing to that of feature type supporting any number of geometry-valued attributes. Indeed the ISO 19117 [2] definition of delineation makes reference to *the spatial attribute that is used to portray a feature*, but it is not clear where/how the aforementioned "attribute" is specified. Instead the ISO 19117 class PF_FeaturePortrayal simple uses a codelist-valued attribute to indicate the delineation.

3.4 Alignment with ISO 19117

ISO 19117 provides a conceptual schema for describing symbols and portrayal mappings. The OGC SE constructs map roughly to the ISO classes but there should be better alignment. This task is presumably under review. In addition, ISO 19117:2005 is presently under revision. However, the following have been noted as part of the TENET AML SE study. This review is based on [2].

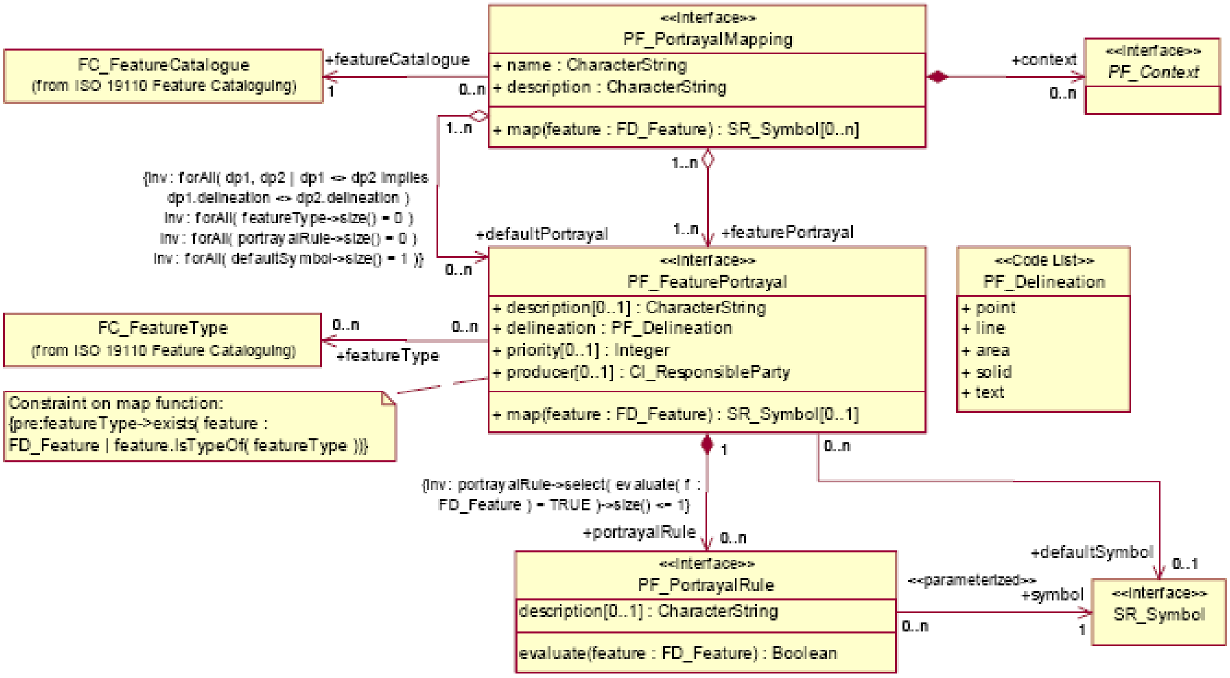


Figure 12: Mapping Components of ISO 19117 Portrayal Mapping Package

OGC SE FeatureTypeStyle corresponds, broadly, to the ISO 19117 interface class PF_FeaturePortrayal, Rule to PF_PortrayalRule and Symbolizer to SR_SymbolElement. A detailed analysis reveals several structural and semantic differences. To summarise a few structural differences in these concepts:

- PF_FeaturePortrayal uses an attribute to specifically identify the geometry delineation; FeatureTypeStyle does not explicitly support this concept.
- ISO 19117 models portrayal mapping conditions such as scale, lighting and display medium using the interface PF_Context attached to a higher level interface PF_PortrayalMapping; SE supports scale conditions at the level of the FeatureTypeStyle element.
- A PF_PortrayalRule instance is associated with a single SR_Symbol – a composition of other subordinate SR_Symbols and leaf SR_SymbolElements; whereas an SE Rule element may consist of multiple Symbolizers but these do not support the hierarchical structure of SR_Symbol.