Open Geospatial Consortium

Date: 2011-11-23

Reference number of this document: OGC 11-089r1

Category: Public Engineering Report

Editor: Daniel Tagesson

OGC® OWS-8 Engineering Report - Guidelines for International Civil Aviation Organization (ICAO) portrayal using SLD/SE

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

.

Document type:	OpenGIS® Public Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

Preface

This document is a deliverable for the OGC Web Services 8 (OWS-8) testbed activity. OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate standards or revisions to existing standards into OGC's Standards Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

The OWS-8 sponsors are organizations seeking open standards for their interoperability requirements. After analyzing their requirements, the OGC Interoperability Team recommended to the sponsors that the content of the OWS-8 initiative be organized around the following threads:

* Observation Fusion

* Geosynchronization (Gsync)

* Cross-Community Interoperability (CCI)

* Aviation

More information about the OWS-8 testbed can be found at:

http://www.opengeospatial.org/standards/requests/74

OGC Document [11-139] “OWS-8 Summary Report” provides a summary of the OWS-8 testbed and is available for download:

https://portal.opengeospatial.org/files/?artifact_id=46176

License Agreement

Contents

Figures

Page

Tables Page

OGC® OWS-8 Engineering Report - Guidelines for ICAO portrayal using SLD/SE

1 Introduction

1.1 Scope

This OGC® document gives guidelines to portrayal of AIXM according to ICAO aviation symbology using SLD/SE.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Örjan Kvartsberg	Carmenta
Daniel Tagesson	Carmenta
Jeroen Dries	Luciad
Simon Merrick	Envitia
Alex Brooker	Envitia

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2011-05-06	Draft 1	Örjan Kvartsberg	Initial draft	An initial draft of the document. Mainly layout
2011-05-09	Draft 2	Daniel Tagesson	Various	Added general sections on the scope and focus of the document.
2011-05-16	Draft 3	Daniel Tagesson	Various	
2011-06-20	Draft 4	Daniel Tagesson	6 and 7	Updated per feedback from Eurocontrol and added new information from Envitia.
2011-07-27	Draft 5	Daniel Tagesson	Title and headers	Set OGC document number 11-089.
2011-08-27	Draft 6	Daniel Tagesson	Various	Updated to final draft
2011-09-29	Final	Daniel Tagesson	Various	Minor updates for the release version
2011-10-5	Final	Carl Reed	Various	Prepare for publication

1.4 Future work

A number of areas have been identified for future work.

How to support composite graphics, such as that required to properly visualize runway segments. See chapter 7.1.4.

Solutions to the Directional Associations issues as discussed in Annex A.

Further work on composite child symbolizers and composite rules as discussed in Annex B.

Implementing portrayal for ICAO weather symbology according to ICAO Annex 3, using SLD/SE has not been explored and needs to be investigated.

1.5 Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS® Web Services Common Standard*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

ICAO Annex 3 to the Convention on International Civil Aviation, Meteorological Service for International Air Navigation, Edition 17, July 2010

ICAO Annex 4 to the Convention on International Civil Aviation, Aeronautical Charts, Edition 11, July 2009

OGC 11-073 - OWS-8 WFS Guidance for AIXM, September 2011

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

4 Conventions

4.1 Abbreviated terms

AIXM	Aeronautical Information Exchange Model
ATS	Air Traffic Service
FAA	Federal Aviation Administration
FPS	Feature Portrayal Service
GML	Geography Markup Language
ICAO	International Civil Aviation Organization
OGC	Open Geospatial Consortium
OWS	OGC Web Services
SE	Symbology Encoding
SLD	Styled Layer Descriptor
UML	Unified Modelling Language
XML	Extensible Markup Language
XSD	XML Schema

4.2 UML notation

Most diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5 Guidelines for ICAO portrayal using SLD/SE overview

5.1 Focus

The intent of the project has not been to develop a complete SLD for all ICAO styles/symbology specified in ICAO Annexes 3 and 4. Instead, the focus has been on identifying the issues that come up when trying to accommodate for ICAO styling/symbology in SLD, and identifying solutions to do so. So the SLD:s produced focus on a subset of ICAO styles/symbology, and represent recommendations on how to encode those in SLD.

6 Issues

The issues identified as the focus for these guidelines are of three types:

- Cartographic/presentation; limited by what types of styles and symbols are possible to describe using SE.
- Data dependent; which are derived from the complexities of portraying AIXM using SLD/SE.
- Status information, such as closed/unserviceable status of features.

6.1 Cartographic/Presentation limitations

6.1.1 ICAO Airspace band style

Airspaces in ICAO Annex 4 are visualized using inner fills polygons which may be complex to construct.

6.1.2 Special patterns

ICAO Annex 4 contains styles with fill patterns such as checkerboard or slanted lines, which are complex to achieve.

6.1.3 Label deconfliction

Aeronautical data typically contains large amounts of data which are hard or impossible to create a readable visualization from without label deconfliction. Different data may have different importance when doing that so rules, priorities or hints to deconfliction may be necessary to enable such functionality in clients.

6.2 Data dependent

Portrayal of AIXM is complex due to the hierarchical and relational structure of the data.

6.3 Portrayal of status information

Status information, such as closed, contaminated or unserviceable is not covered by the ICAO documentation but typically what you are interested in as a user subscribing to, or viewing AIXM data.

7 Guidelines and solutions to issues

7.1 Cartographic/Presentation limitations

7.1.1 ICAO Airspace band style

Airspaces in ICAO Annex 4 are visualized using inner fills polygons which may be complex to construct. See the image from the ICAO Annex 4 below.

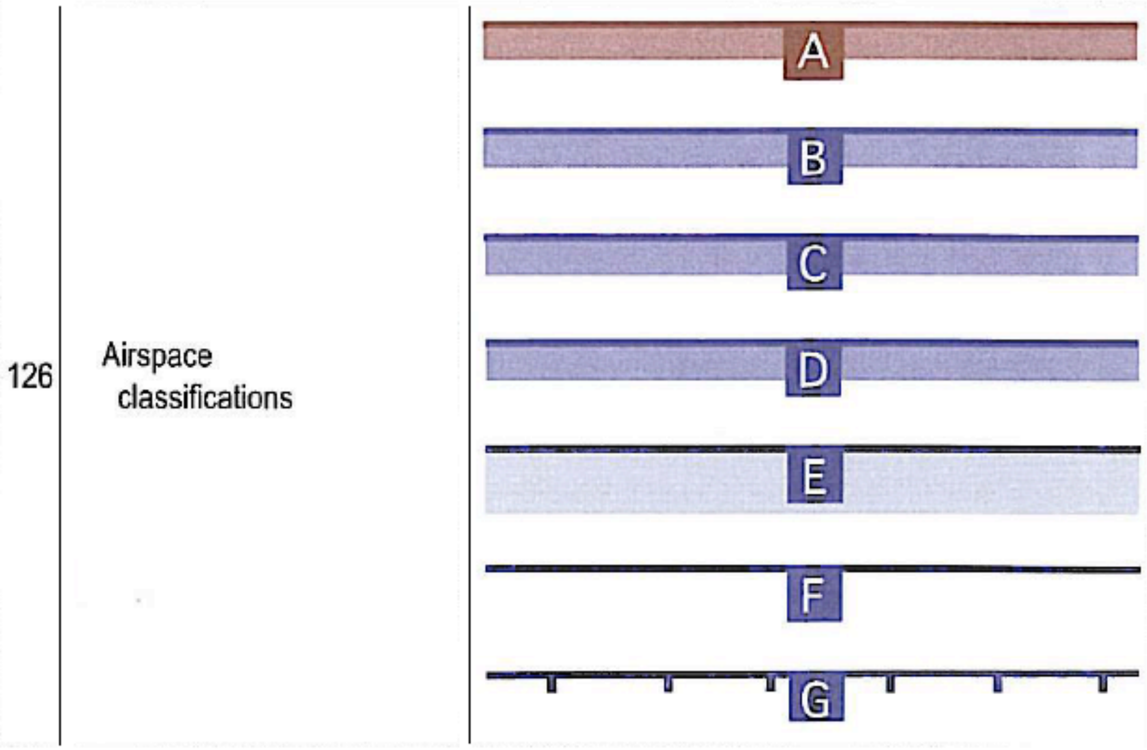


Figure 1 - ICAO Airspace style

7.1.1.1 Problem

Airspaces have relatively complex visualizations and often large polygons with large number of points.

7.1.1.2 Recommendations

7.1.1.2.1 Visualization

Classification A to D could be achieved by the following example. Instead of having two sets of rules with one for classification A and one for B-D, Recode could be used to set the stroke colors.

```
<se:Rule>
  <se:PolygonSymbolizer>
    <se:Stroke>
      <se:SvgParameter name="stroke">#0000ff</se:SvgParameter>
      <se:SvgParameter name="stroke-
      opacity">0.3</se:SvgParameter>
      <se:SvgParameter name="stroke-width">4</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linejoin">round</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linecap">square</se:SvgParameter>
    </se:Stroke>
    <se:PerpendicularOffset>-2</se:PerpendicularOffset>
  </se:PolygonSymbolizer>
  <se:PolygonSymbolizer>
    <se:Stroke>
      <se:SvgParameter name="stroke">#160E5B</se:SvgParameter>
      <se:SvgParameter name="stroke-
      opacity">1.0</se:SvgParameter>
      <se:SvgParameter name="stroke-width">2</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linejoin">round</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linecap">square</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
</se:Rule>
```

Classification E can be done in the same way but with a wider and more transparent first stroke and Classification F would be visualized without the transparent stroke altogether.

Classification G can be visualized by using a Stroke with "stroke-dasharray" set to a appropriate value and combine it with "stroke-width" to set the "length" of the line. Then apply a PerpendicularOffset to move the dashes.

```
<se:Rule>
  <se:PolygonSymbolizer>
    <se:Stroke>
      <se:SvgParameter name="stroke">Black</se:SvgParameter>
      <se:SvgParameter name="stroke-
      opacity">1.0</se:SvgParameter>
      <se:SvgParameter name="stroke-width">2</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linejoin">round</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linecap">square</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
  <se:PolygonSymbolizer>
    <se:Stroke>
      <se:SvgParameter name="stroke">Black</se:SvgParameter>
      <se:SvgParameter name="stroke-
      opacity">1.0</se:SvgParameter>
      <se:SvgParameter name="stroke-width">5</se:SvgParameter>
      <se:SvgParameter name="stroke-
      linecap">square</se:SvgParameter>
      <se:SvgParameter name="stroke-
     dasharray">1,15</se:SvgParameter>
    </se:Stroke>
    <se:PerpendicularOffset>-2</se:PerpendicularOffset>
  </se:PolygonSymbolizer>
</se:Rule>
```

7.1.1.2.2 Geometrical complexity

Airspaces are often large polygons, with large number of points, for example when following country borders. This causes large volumes of data to transfer and heavy processing when portraying the information. To address this issue, the geometries would have to be simplified with level-of-detail support, as discussed further in chapter 7.4.2.

7.1.2 Special patterns

7.1.2.1 Problem

ICAO Annex 4 contains styles with fill patterns such as checkerboard or slanted lines, which are complex to achieve.

7.1.2.2 Recommendations

7.1.2.2.1 GraphicFill/Stroke

The majority of the complex fill patterns in the ICAO guidelines can simply be defined using raster image tiles in a format that supports transparency. Encoding in SLDs can then easily be achieved using the GraphicFill and GraphicStroke elements from the SE 1.1 specification.

This approach has been validated using Closed Surfaces (Runways) as the case study, see chapter 7.3.1. The validation was performed using two separate FPSs.

7.1.2.2.2 Complex Fills

The following SLD snippet shows the simple encoding that has been used by both implementations to render special patterns to fill area geometry.

```
<se:PolygonSymbolizer>
  <se:Geometry>
    <ogc:PropertyName>...</ogc:PropertyName>
  </se:Geometry>
  <se:Fill>
    <se:GraphicFill>
      <se:Graphic>
        <se:ExternalGraphic>
          <se:OnlineResource xlink:href="http://localhost/images/Fills/red_diag.png" xlink:type="simple"/>
          <se:Format>image/png</se:Format>
        </se:ExternalGraphic>
      </se:Graphic>
    </se:GraphicFill>
  </se:Fill>
```

```
</se:PolygonSymbolizer>
```

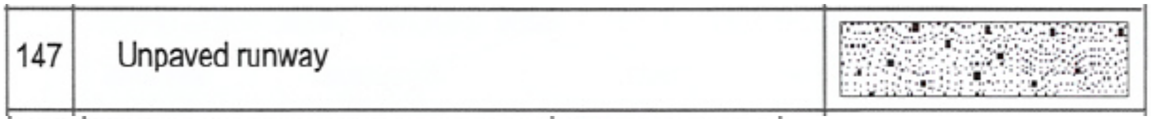
7.1.2.2.3 Complex Lines

The following SLD snippet shows the simple encoding that has been used by both implementations to render special patterns to fill area geometry. The Graphic Stroke can be used in this way for line geometry, or for the borders of polygon geometry.

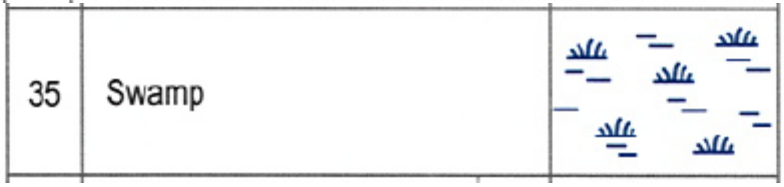
```
<sld:LineSymbolizer>
  <sld:Geometry>
    <ogc:PropertyName>GEOMETRY</ogc:PropertyName>
  </sld:Geometry>
  <sld:Stroke>
    <sld:GraphicStroke>
      <sld:Graphic>
        <sld:ExternalGraphic>
          <sld:OnlineResource xlink:href="http://localhost/images/Lines/0716b_magenta.png"
xlink:type="simple"/>
          <sld:Format>image/png</sld:Format>
        </sld:ExternalGraphic>
        <!-- sets the line width to be 8 pixels, which scales up / down the png file for rendering -->
        <sld:Size>8</sld:Size>
      </sld:Graphic>
    </sld:GraphicStroke>
  </sld:Stroke>
</sld:LineSymbolizer>
```

7.1.2.2.4 GraphicFill/Stroke versus "Pen-Up, Pen-Down"

The obvious advantage of the GraphicFill approach is that arbitrarily complex fill patterns can be easily encoded in SLDs and rendered by FPS implementers. This avoids the need to define the required mechanisms in the SE specification that would allow the definition of arbitrarily complex fill patterns. Such a mechanism would potentially need to extend to the pen-up/pen-down level of expressive capability (consider the fill style required for Unpaved Runway [147], Swamp [35], or Rice field [36]).



147	Unpaved runway	



35	Swamp	

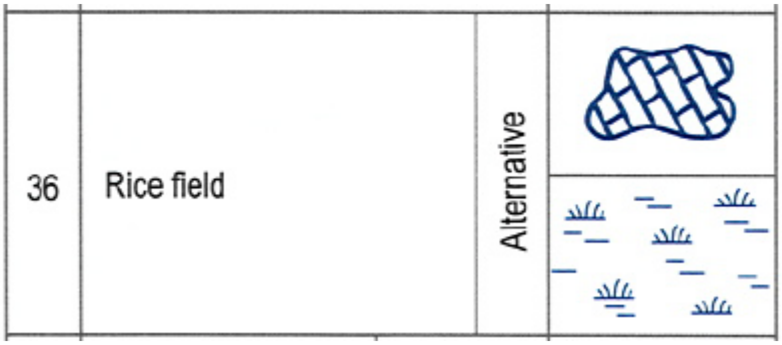


Figure 2 - ICAO fill style examples

The expressive capability required would not only be a far greater overhead for implementers, with all of the implied issues (limitations of vendor-specific implementations, misinterpretations of the specification etc.) but would result in the potential for very verbose SLDs. This in itself represents performance implications (network traffic / bandwidth) for heavily used systems.

The obvious disadvantage is that for simpler fill styles, a vector rendering would be much more efficient and hence higher performance could be achieved by the FPS. The hatched pattern suggested for SE 1.2 is a good example.

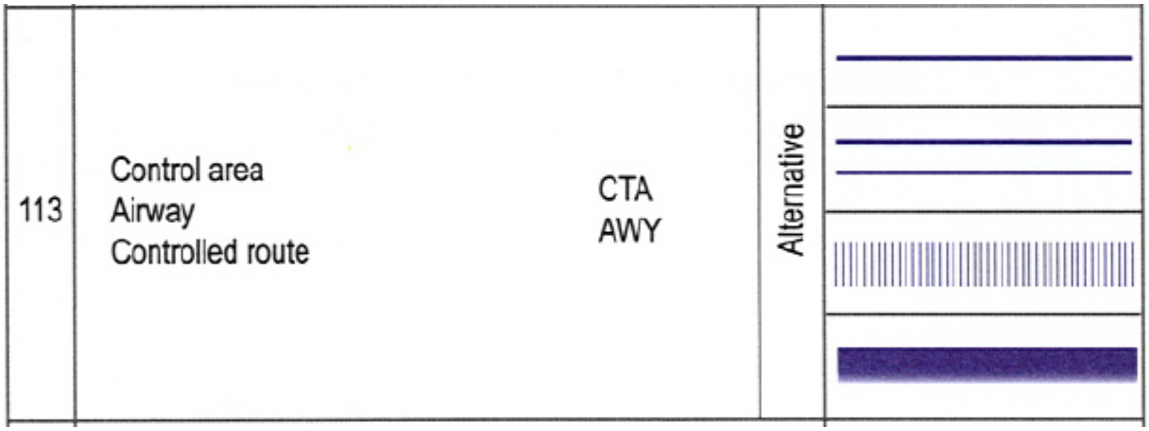
However, one of the FPSs used to test this approach applies an interesting optimization in this area that could be used by other FPS implementations, particularly when working in a fixed enterprise/domain such as AIM: The FPS can be pre-configured to pair up the URLs of external graphics with well known rendering styles (points / lines / fills).

This optimization enables the software to perform its actual rendering task using its intrinsically high performance vector rendering (in place of the more cumbersome raster tiling). This would allow vendors to optimize their FPS implementations for well known domains, and yet would allow the enterprise community to rely only on the standards when choosing their service providers.

Note: both techniques were shown to work using different FPS software, although no performance testing was undertaken to quantify the performances of the two approaches.

7.1.2.2.5 Translucency within complex line styles

In parallel with the discussion above on the benefits of GraphicFill over a full Pen-up/Pen-down expressive capability, is another example from Aeronautical Charts. This example is a line style that seems to incorporate a gradient of translucency. The final "alternative" style for Control area Airway Controlled route features (see image below) appears to vary translucency within the linestyle itself. If this is the case, or if line styles of this nature need to be portrayed by SLDs / FPS services, then this represents an extra layer of complexity for the potential pen-up / pen-down solution and yet this could easily be achieved with the GraphicFill approach without extension to the SE specification.



113	Control area Airway Controlled route	CTA AWY	Alternative

Figure 3 - ICAO ATS line styles

7.1.2.2.6 GraphicFill using inline SVG

Related to the use of a GraphicFill? to achieve complex patterns, it should also be noted that it is possible to use an embedded SVG image. This avoids having external dependencies in your SLD which makes it easier to exchange. The use of SVG also makes the resulting SLD more readible/editable, as opposed to embedding an BASE64 encoded image.

Below is an example of a RunwayElement? feature type style with an embedded SVG that defines a checkerboard pattern. Note that the SVG can probably be optimized a bit further to reduce its size.

```
<?xml version='1.0' encoding='UTF-8'?>
<se:FeatureTypeStyle xmlns:se="http://www.opengis.net/se" xmlns:o
gc="http://www.opengis.net/ogc" xmlns:xsd="http://www.w3.org/2001
/XMLSchema" xmlns:sld="http://www.opengis.net/sld" xmlns:gml="htt
p://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlin
k" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/ogc http://s
chemas.opengis.net/filter/1.1.0/filter.xsd http://www.w3.org/2001
/XMLSchema http://www.w3.org/2001/XMLSchema.xsd http://www.opengi
s.net/se http://schemas.opengis.net/se/1.1.0/FeatureStyle.xsd htt
p://www.opengis.net/sld http://schemas.opengis.net/sld/1.0.0/Styl
edLayerDescriptor.xsd http://www.opengis.net/gml http://schemas.o
pengis.net/gml/3.1.1/base/gml.xsd http://www.w3.org/1999/xlink ht
tp://schemas.opengis.net/xlink/1.0.0/xlinks.xsd " version="1.1.0"
>
  <se:Name>RunwayElementType</se:Name>
  <se:FeatureTypeName>RunwayElementType</se:FeatureTypeName>
  <se:Rule>
    <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
    <se:MaxScaleDenominator>INF</se:MaxScaleDenominator>
    <se:PolygonSymbolizer>
      <se:Fill>
        <se:GraphicFill>
          <se:Graphic>
            <se:ExternalGraphic>
```

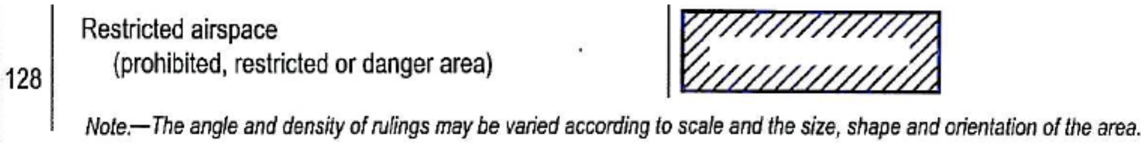
```
              <se:InlineContent encoding="xml">
                <svg xmlns="http://www.w3.org/2000/svg" xmlns:cc=
"http://creativecommons.org/ns#" xmlns:sodipodi="http://sodipodi.
sourceforge.net/DTD/sodipodi-
0.dtd" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
 xmlns:svg="http://www.w3.org/2000/svg" xmlns:dc="http://purl.org
/dc/elements/1.1/" contentScriptType="text/ecmascript" zoomAndPan
="magnify" contentStyleType="text/css" id="svg2" sodipodi:docname
="checkerboard2.svg" version="1.1" width="32" preserveAspectRatio
="xMidYMid meet" height="32">
                  <sodipodi:namedview objecttolerance="10" border
color="#666666" inkscape:window-
height="968" inkscape:zoom="9.90625" gridtolerance="10" id="named
view11" inkscape:current-layer="svg2" inkscape:window-y="-
3" inkscape:cy="17.211356" inkscape:window-x="-
2" inkscape:cx="0.15141956" showgrid="false" guidetolerance="10"
inkscape:pageopacity="0" inkscape:pageshadow="2" inkscape:window-
width="1280" pagecolor="#ffffff" borderopacity="1" inkscape:windo
w-maximized="1"/>
                  <defs id="defs4">
                    <pattern xlink:href="#Checkerboard" xlink:typ
e="simple" xlink:actuate="onRequest" patternTransform="matrix(7.6
424715,0,0,7.7030688,4.6949061,239.26733)" id="pattern5237" prese
rveAspectRatio="xMidYMid meet" xlink:show="replace"/>
                    <pattern width="2" patternUnits="userSpaceOnU
se" xlink:type="simple" xlink:actuate="onRequest" height="2" patt
ernTransform="translate(0,0) scale(10,10)" id="Checkerboard" pres
erveAspectRatio="xMidYMid meet" xlink:show="replace">
                      <rect x="0" y="0" width="1" style="fill:bla
ck;stroke:none" height="1" id="rect4549"/>
                      <rect x="1" y="1" width="1" style="fill:bla
ck;stroke:none" height="1" id="rect4551"/>
                    </pattern>
                  </defs>
                  <g id="layer1" style="stroke:none" transform="t
ranslate(0,-1020.3622)">
                    <rect x="0.030934934" y="1020.5137" width="32
.241737" style="fill:url(#pattern5237);fill-opacity:1;fill-
rule:nonzero;stroke:none;stroke-width:0.76727098000000005;stroke-
miterlimit:4;stroke-
dasharray:0.76727100999999998, 0.76727100999999998;stroke-
dashoffset:0" rx="0" height="32.156994" ry="0" id="rect5198"/>
                  </g>
                </svg>
              </se:InlineContent>
              <se:Format>image/svg+xml</se:Format>
            </se:ExternalGraphic>
          </se:Graphic>
        </se:GraphicFill>
      </se:Fill>
      <se:Stroke>
        <se:SvgParameter name="stroke-opacity">
```

```
          <ogc:Literal>1.0</ogc:Literal>
        </se:SvgParameter>
        <se:SvgParameter name="stroke">#000000</se:SvgParameter>
        <se:SvgParameter name="stroke-width">
          <ogc:Literal>1</ogc:Literal>
        </se:SvgParameter>
      </se:Stroke>
    </se:PolygonSymbolizer>
  </se:Rule>
</se:FeatureTypeStyle>
```

7.1.3 ICAO Restricted airspace styling

7.1.3.1 Problem

Restricted airspaces have a styling which is very complex. See the imge from the ICAO Annex 4 below.



128 Restricted airspace
(prohibited, restricted or danger area)

Note.—The angle and density of rulings may be varied according to scale and the size, shape and orientation of the area.

Figure 4 - ICAO Restricted airspace styling

7.1.3.2 Recommendation

The recommended style is not possible to achieve in an SLD today. Technically it would require support for using a GraphicFill as mentioned above, but only within a bufferzone inside a polygon.

Repeating a symbol along the line is another way to achive a similar presentation, however this causes issues primarily in angles, where the visualization will get cluttered. An example of some of the problems can be seen below.

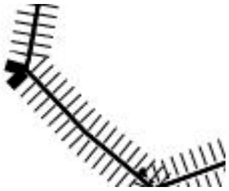


Figure 5 - Line with repeated symbol

The recommendation is therefore to use visualization such as that for controlled airspaces with a semitransparent inner line or a polygon with a semitransparent inner fill. Below is an example of the former. The visualization should use a distinct red color to highlight the area.

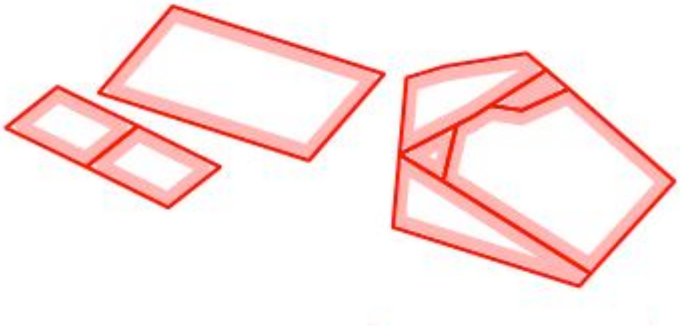


Figure 6 - Restricted airspaces alternate visualization

7.1.4 Airways

7.1.4.1 Problem

The symbol for enroute airways in ICAO Annex 4 is simply a black line, but Aneex 4 chapter 7.9.3.1.1 the required components for ATS are listed. For ATS routes in en-route flight for example designators and the track in both directions should be displayed, together with minimum heights and distances between significant or reporting points.

7.1.4.2 Recommendations

Here are two examples of how that information could be visualized.

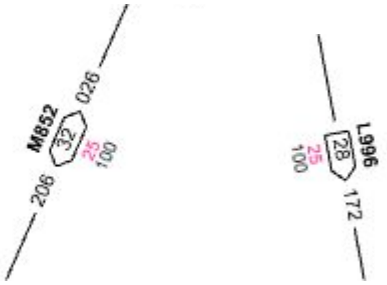


Figure 7 - Airway symbol example

The first image shows an example with two directions and the second an airway with one direction.

With SE 1.1.0 this is be hard to accomplish. In SE 1.2 there are some proposed changes that probably could be used. A CompositeGraphic used in a GraphicStroke could accomplish some of the desired visualization and with RelativeOrientation set to "line" the symbol would be rotated along the line.

What remains is that the line should be clipped by the bound box of the CompositeGraphic. Not sure if the sentence means that a gap could be created by setting the length attribute to a value "Length gives the linear length of the geometry to reserve for the graphic icon and is allowed to be zero, which means not to advance the linear plotting position before or after drawing the graphic.". For a CompositeGraphic the resulting length is hard to know, wouldn't it be better to have an element "ClipLine" which tells that the line should be clipped by the graphics bounding box. A Padding or gap element is probably needed as well to be able to extend the clip area. This approach makes the configuration simpler but perhaps it requires a more complex implementation.

An alternative could be to use the CompoundStroke. The use of PreGap and PostGap seems a little awkward but perhaps I'm only misinterpreting it. How can PreGap and PostGap be used in CompoundStroke to create a gap? Explanatory text from the document "The PreGap tells how far to advance along the line before starting to plot content and the PostGap tells how far from the end of the line to stop all plotting."

7.1.5 Label deconfliction

7.1.5.1 Problem

Aeronautical data typically contains large amounts of data which are likely to be hard or impossible to create a readable visualization from without label deconfliction. Different data may have different importance when doing that so rules, priorities or hints to deconfliction may be necessary to enable such functionality in clients.

7.1.5.2 Recommendations

Label deconfliction is not supported in SE and likely far too complex to support in any detail. It should rather be up to the clients to support this, but in order to do so the client needs to know what it can and cannot do. Typically in regards to what features can be hidden or how much a label can be displaced and still create a good enough result.

To support this deconfliction attributes will be necessary in SLD. There is a change request regarding this issue CR 11-023 but it is only for text labels. More complex symbols like a CompositeGraphic could also need deconfliction and organizing and an additional change proposal is necessary.

The proposal is to add a new element Deconfliction. The element CanBeRemoved in Deconfliction determines if the symbol can be removed if it cannot be placed without overlapping another symbol in the same layer. The element Priority should work as in CR 11-023 and is used to set importance of a symbol. The priority could be set with a number directly or by fetching a value from a property.

```
<xsd:element name="Deconfliction">
 <xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="se:CanBeRemoved" minOccurs="0"/>
    <xsd:element ref="se:Priority" minOccurs="0"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:element>

<xsd:element name="CanBeRemoved" type="xsd:boolean"/>
<xsd:element name="Priority" type="sld:ParameterValueType"/>
```

```
The possibility to use deconfliction is needed on
PointSymbolizer? and TextSymbolizer? and they should be modified
to look like this.
<xsd:element name="PointSymbolizer"
substitutionGroup="se:Symbolizer">
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="se:SymbolizerType">
    <xsd:sequence>
     <xsd:element ref="se:Geometry" minOccurs="0"/>
     <xsd:element ref="se:UnitOfMeasure" minOccurs="0"/>
     <xsd:element ref="se:Transform" minOccurs="0"/>
     <xsd:element ref="se:Graphic"/>
     <xsd:element ref="se:Deconfliction" minOccurs="0"/>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="TextSymbolizer"
substitutionGroup="se:Symbolizer">
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="se:SymbolizerType">
    <xsd:sequence>
     <xsd:element ref="se:Geometry" minOccurs="0"/>
     <xsd:element ref="se:UnitOfMeasure" minOccurs="0"/>
     <xsd:element ref="se:PerpendicularOffset" minOccurs="0"/>
     <xsd:element ref="se:Transform" minOccurs="0"/>
     <xsd:element ref="se:Label" minOccurs="0"/>
     <xsd:element ref="se:Deconfliction" minOccurs="0"/>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:element>
```

The current suggestion of the Deconfliction element is a bare minimum and it probably needs to be extended to give more control over the deconflition behavior. Perhaps the ability to specify a deconfliction algorithm (e.g. 'explode on click') would be useful. A code-list of well known deconfliction algorithms could be identified.

7.2 Data dependent

Portrayal of AIXM is complex to achieve using SLD and SE due to the hierarchical and relational structure of the data.

7.2.1 Specifying geometries in symbolizers

7.2.1.1 Problem

Symbolizers use xpath expressions to point to the geometry that they want to display.

7.2.1.2 Recommendations

Different implementers tend to support different xpaths in this element, so we need to agree on a few best practices:

- Do not specify a geometry in case the feature only has one tag in which the geometry can be defined, this allows a bit more freedom for the implementation, which can be useful for instance in airspaces. (Airspaces can have multiple geometry components that need to be combined.)

- Always point to the actual geometry, not to the property around it. For instance AirportHeliportTimeSlice /ARP/ElevatedPoint, not AirportHeliportTimeSlice /ARP.

- Avoid the use of '//', even though this is a convenient shortcut, since it does force the implementation to do a search through the entire feature.

7.2.2 Portrayal of hierarchical data

7.2.2.1 Problem

The AIXM model contains many feature types of a hierarchical nature. There is a need to encode styles for such features using rule conditions that evaluate the properties of the features' nested children.

7.2.2.2 Recommendations

The AIXM model contains many feature types of a hierarchical nature. There is a need to encode styles for such features using rule conditions that evaluate the properties of the features' nested children.

A change request entitled *SE Symbolizer and Rule for styling of nested child objects* has been raised to allow different style to be applied to nested children based on the childrens' properties. See Annex C for more details on the change request.

This additional flexibility is an important tool for the styling and portrayal of AIXM data and feeds into the following section on the portrayal of relational data.

7.2.3 Portrayal of relational data

7.2.3.1 Problem

Many of AIXM's feature types are conceptually composite types. The AIXM model tends to specify uni-directional relationships between features and their component features. Unfortunately the navigable direction of such relationships also tends to be in the opposite direction to that required for portrayal.

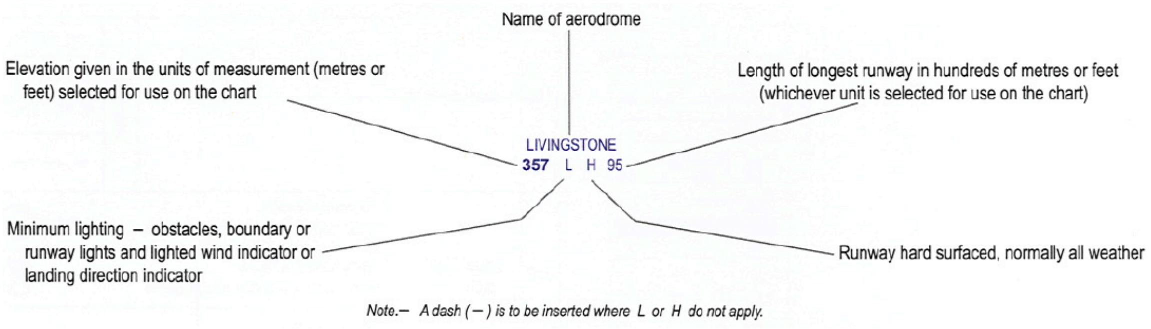
7.2.3.2 Recommendations

In order to portray AIXM data such as the examples below, it is recommended that the WFS services behind the FPS should be either an Xlink WFS (WFS 1.1.0) or be WFS 2.0.0 compliant.

This recommendation and others are explored in the airport portrayal example below and in Annex A.

7.2.3.3 Example: Airport Portrayal

The Aeronautical Charts (Annex 4 to the Convention on International Civil Aviation) calls for an abbreviated form which may be in association with Aerodrome symbols. This abbreviated form looks like this:


Name of aerodrome
Elevation given in the units of measurement (metres or feet) selected for use on the chart
Length of longest runway in hundreds of metres or feet (whichever unit is selected for use on the chart)
LIVINGSTONE
357 L H 95
Minimum lighting – obstacles, boundary or runway lights and lighted wind indicator or landing direction indicator
Runway hard surfaced, normally all weather

Note.— A dash (—) is to be inserted where L or H do not apply.

The labels displayed rely on properties of the airport's components, for example the length of the longest runway must be obtained by evaluating all of the airport's runways.

The problem is that an SLD cannot define a rule that can access the associated runways in terms of the airport's properties. In short, the Airport does not contain a reference to its runways, so the SLD cannot access the "length". We can see that the required links are missing from the Airport by inspection of the example GML Airport feature below.

```
<aixm:AirportHeliport gml:id="gmlID102079" xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2">
  <gml:identifier codeSpace="http://www.comsoft.aero/cadas-aimdb/caw">000AB061-864F-4E7C-94B6-
D769C7238AC2</gml:identifier>
  <aixm:timeSlice>
    <aixm:AirportHeliportTimeSlice gml:id="gmlID102084">
    <gml:validTime>
      <gml:TimePeriod gml:id="gmlID102142">
        <gml:beginPosition>2000-01-01T00:00:00.000Z</gml:beginPosition>
        <gml:endPosition>9999-12-31T23:59:59.000Z</gml:endPosition>
      </gml:TimePeriod>
    </gml:validTime>
    <aixm:interpretation>BASELINE</aixm:interpretation>
    <aixm:sequenceNumber>1</aixm:sequenceNumber>
    <aixm:correctionNumber>0</aixm:correctionNumber>
    <aixm:featureLifetime>
      <gml:TimePeriod gml:id="gmlID102143">
        <gml:beginPosition>2000-01-01T00:00:00.000Z</gml:beginPosition>
        <gml:endPosition>9999-12-31T23:59:59.000Z</gml:endPosition>
      </gml:TimePeriod>
    </aixm:featureLifetime>
    <aixm:designator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:name>CURTIS MUNI</aixm:name>
    <aixm:locationIndicatorICAO xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:designatorIATA xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:type>AH</aixm:type>
    <aixm:certifiedICAO xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:privateUse>NO</aixm:privateUse>
    <aixm:controlType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:fieldElevation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:fieldElevationAccuracy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:verticalDatum xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:magneticVariation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:magneticVariationAccuracy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:dateMagneticVariation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:magneticVariationChange xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:referenceTemperature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:altimeterCheckLocation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:secondaryPowerSupply xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:windDirectionIndicator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:landingDirectionIndicator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:transitionAltitude xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:transitionLevel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:lowestTemperature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:abandoned xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:certificationDate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:certificationExpirationDate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <aixm:contaminant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <aixm:servedCity>
      <aixm:City gml:id="gmlID102144">
        <aixm:name>CURTIS</aixm:name>
    </aixm:City>
   </aixm:servedCity>
   <aixm:responsibleOrganisation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
   <aixm:ARP>
    <aixm:ElevatedPoint gml:id="gmlID102145" srsName="urn:ogc:def:crs:OGC:1.3:CRS84">
     <gml:pos>-100.47275 40.63875</gml:pos>
    </aixm:ElevatedPoint>
   </aixm:ARP>
   <aixm:aviationBoundary xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
   <aixm:altimeterSource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
   <aixm:contact xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
   <aixm:availability>
    <aixm:AirportHeliportAvailability gml:id="gmlID102152">
     <aixm:operationalStatus>NORMAL</aixm:operationalStatus>
     <aixm:usage>
      <aixm:AirportHeliportUsage gml:id="gmlID102153">
       <aixm:selection>
        <aixm:ConditionCombination gml:id="gmlID102154">
         <aixm:aircraft>
          <aixm:AircraftCharacteristic gml:id="gmlID102155">
           <aixm:type>LANDPLANE</aixm:type>
          </aixm:AircraftCharacteristic>
         </aixm:aircraft>
        </aixm:ConditionCombination>
       </aixm:selection>
      </aixm:AirportHeliportUsage>
     </aixm:usage>
    </aixm:AirportHeliportAvailability>
   </aixm:availability>
   <aixm:annotation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
  </aixm:AirportHeliportTimeSlice>
 </aixm:timeSlice>
</aixm:AirportHeliport>
```

When we inspect an example of a Runway, we can see the link to the airport feature:

```
<aixm:Runway gml:id="gmlID67205" xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2">
    <gml:identifier codeSpace="http://www.comsoft.aero/cadas-aimdb/caw">00AB92E2-9B9B-48F7-
AD0A-79FD7639F4A5</gml:identifier>
    <aixm:timeSlice>
        <aixm:RunwayTimeSlice gml:id="gmlID67206">
            <gml:validTime>
                <gml:TimePeriod gml:id="gmlID67207">
                    <gml:beginPosition>2000-01-01T00:00:00.000Z</gml:beginPosition>
                    <gml:endPosition>9999-12-31T23:59:59.000Z</gml:endPosition>
                </gml:TimePeriod>
            </gml:validTime>
            <aixm:interpretation>BASELINE</aixm:interpretation>
            <aixm:sequenceNumber>1</aixm:sequenceNumber>
            <aixm:correctionNumber>0</aixm:correctionNumber>
            <aixm:featureLifetime>
                <gml:TimePeriod gml:id="gmlID67208">
                    <gml:beginPosition>2000-01-01T00:00:00.000Z</gml:beginPosition>
                    <gml:endPosition>9999-12-31T23:59:59.000Z</gml:endPosition>
                </gml:TimePeriod>
```

```
                    </aixm:featureLifetime>
                    <aixm:designator>17/35</aixm:designator>
                    <aixm:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
                    <aixm:nominalLength xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:lengthAccuracy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:nominalWidth xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:widthAccuracy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:widthShoulder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:lengthStrip uom="M">2255.8248</aixm:lengthStrip>
                    <aixm:widthStrip uom="M">30.48</aixm:widthStrip>
                    <aixm:lengthOffset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:widthOffset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:abandoned xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:surfaceProperties xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:associatedAirportHeliport xlink:href="http://www.comsoft.aero/cadas-
aimdb/generated#xpointer(//AirportHeliport[gml:identifier=000AB061-864F-4E7C-94B6-
D769C7238AC2'])" xmlns:xlink="http://www.w3.org/1999/xlink"/>
                    <aixm:overallContaminant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:annotation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                    <aixm:areaContaminant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
              </aixm:RunwayTimeSlice>
       </aixm:timeSlice>
</aixm:Runway>
```

In order to portray AIXM data such as the examples above, it is recommended that the WFS services behind the FPS should be either an Xlink WFS (WFS 1.1.0) or be WFS 2.0.0 compliant. The onus will then be on the WFS (with appropriate instruction from the FPS regarding Xlink resolution) to ensure that the results of a query include not only the requested features, but any related features that are referenced (xlinked) by the requested features, which as mentioned above may well be required / used by the SLD.

For example with a WFS 2.0.0 compliant WFS, we can use the resolve and resolveDepth parameters of the GetFeature query to ensure that all of the GML content required for portrayal of this airport and runway length is included in the result:

```
<wfs:GetFeature
service="WFS"
version="2.0.0"
resolveDepth="1"
resolveTimeout="1"
outputFormat="application/gml+xml; version=3.2"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
```

```
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:Query typeNames="Runway">
    <wfs:PropertyName>aixm:timeSlice/aixm:RunwayTimeSlice/aixm:lengthStrip</wfs:PropertyName>
    <wfs:PropertyName resolve="all"
resolveDepth="1">aixm:timeSlice/aixm:RunwayTimeSlice/aixm:associatedAirportHeliport</wfs:Property
Name>
    <fes:Filter>
      <fes:ResourceId rid="gmlID67205"/>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

The above query yields the following result with the airport definition included inline:

```
<aixm:Runway gml:id="gmlID67205" xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2">
        <aixm:timeSlice>
                <aixm:RunwayTimeSlice gml:id="gmlID67206">
                        <aixm:lengthStrip uom="M">2255.8248</aixm:lengthStrip>
                        <aixm:associatedAirportHeliport>
                                <aixm:AirportHeliport gml:id="gmlID102079"
xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2">
                                        <gml:identifier codeSpace="http://www.comsoft.aero/cadas-
aimdb/caw">000AB061-864F-4E7C-94B6-D769C7238AC2</gml:identifier>
                                        <!-- contents omitted for brevity-->
                                </aixm:AirportHeliport>
                        </aixm:associatedAirportHeliport>
                </aixm:RunwayTimeSlice>
        </aixm:timeSlice>
</aixm:Runway>
```

Using the CompositeChildSymbolizer, this allows an SLD to render the airport geometry as well as a label containing the runway length. There is one remaining problem with this however in that the SLD would need to define a FeatureTypeStyle for the RunwayElement to achieve this portrayal.

Logically speaking the composition is the other way around; the Runway forms part of the Airport, not vice versa. But more importantly, the style may need to account for more than just the component Runways of an Airport; it may wish to create rules based on the properties of an associated Apron or RadarSystem.

In order for a WFS to collect the required features, the associations need to be reversed. If the Airport held xlink properties to its Runways, Aprons and RadarSystems, then the FeatureTypeStyle could target the Airport itself. This not only makes more semantic sense for system developers and administrators, but enables the required flexibility for the portrayal of complex composite features as found in AIXM.

Therefore, it is further recommended that the "reverse associations" extension of AIXM 5.1 be employed for the purposes of defining AIXM portrayal SLDs.

As a final comment, future versions of the AIXM UML Application Schema model may want to consider specifying bi-directional associations where applicable in the aerospace domain. Note that this comment applies to the AIXM conceptual model (i.e. Platform independent UML model). The two-way navigability of the relationships could then be reflected in derived "encoding models" such as the GML application schema. The encoding model for data storage, in a database for example, would not need to be affected.

See also Annex A.

7.2.4 AIXM Temporality

7.2.4.1 Problem

The temporality model of AIXM 5 is a problem, since it would typically require filtering of features based on temporal properties.

7.2.4.2 Recommendations

The temporality aspect has not been explored to any great depths within the portrayal project but a few recommendations can still be pointed out:

Some temporal filtering can be performed from within an SLD document but should preferably be performed at the WFS-level, to greatly simplify the SLDs, reduce the workload for the portrayal service and to reduce the amount of data transferred between the data provider and portrayal service. Querying a WFS for AIXM-data is covered in further detail in *OGC 11-073 - OWS-8 WFS Guidance for AIXM.*

It is however possible to perform time dependent styling as explored in the SAA-pilot project. This showed that a requirement for time dependent styling is the need for an evaluateDuring function, which is also covered in the WFS guidance document.

7.3 Portrayal of status information

7.3.1 Closed/unserviceable features

7.3.1.1 Problem

The ICAO Annex 4 contains no recommendation on how to portray availability information such as closed runways and taxiways, unserviceable navaids etc.

7.3.1.2 Recommendations – Surface Availability (Runways/Taxiways)

We propose the use of an overlaid style for indicating status of a surface; a solid red fill for closed surfaces and an outlined hatched fill for non-normal statuses. The outline is especially important when viewing data at higher zoom levels as a hatching pattern becomes very indistinct as can be seen in Figure 9 and Figure 10.

The status of the surface will be retrieved from the availability element using the following XPath: aixm:timeSlice/aixm:RunwayElementTimeSlice/aixm:availability/aixm:ManoeuvringAreaAvailability/aixm:operationalStatus and the constant can have four different values: 'NORMAL','LIMITED', 'CLOSED' and 'OTHER'.

We suggest that surfaces with status 'CLOSED' be visualized with a solid red fill and the 'LIMITED' and 'OTHER' availability, be visualized with the hatched pattern in amber and yellow respectively as to avoid the stronger danger indication that red provides. Pros of using a solid color is of course that it is a simpler visualization and red will make it very clear that the surface is unavailable. A con of using colors as indicators is that they make it hard to distinguish between open and closed surfaces in black and white.

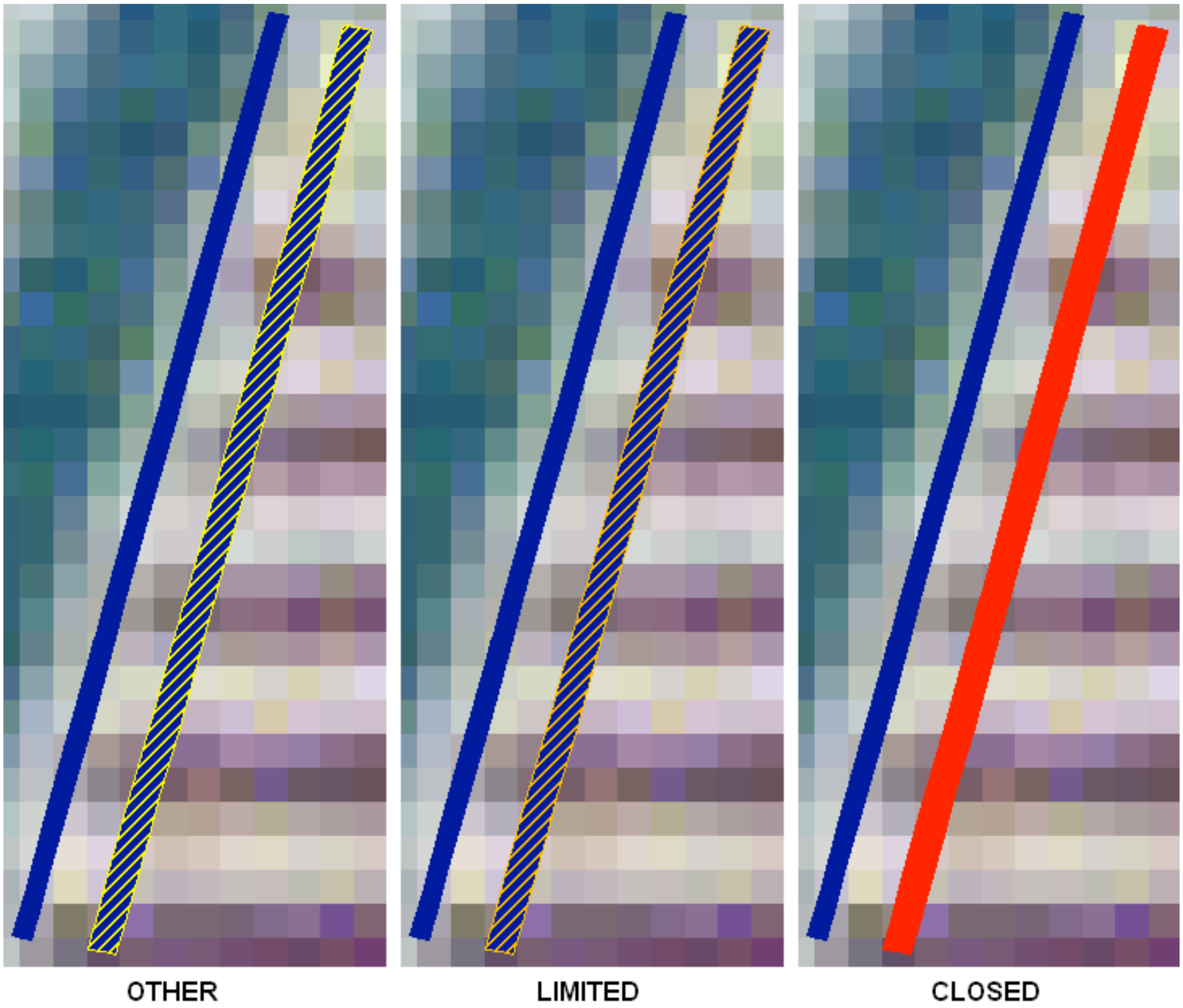


Figure 8 - Surface status visualization example

7.3.1.2.1 Hatching pattern

The hatched fill pattern can be achieved using raster image tiles in a format that supports transparency. Encoding in SLDs can then easily be achieved using the GraphicFill? element from the SE 1.1 specification.

Optionally a HatchedFill keyword is being considered for SE 1.2. According to that schema, a hatching fill would look like:

```
<HatchedFill uom="urn:ogc:def:uom:se::px">
  <Stroke>SE stroke parameters such as color and line width</Stroke>
  <Angle>45</Angle>
  <!--Default is 45 so this can be dropped-->
  <Distance>8</Distance>
  <!--Distance in pixels, should we use meters?-->
</HatchedFill>
```

One minor problem with a hatching pattern is that viewing a runway with bearing 45/225 with north up would not visualize the status as well, since the hatch pattern would be parallel with the runway. Ideal would be a hatch angle which is relative to the runway bearing. But unfortunately, the runway elements themselves do not contain this information. It is available in the runway direction, but there is no clear path from the elements to the direction. For further discussion on this type of problems, see Annex A.

7.3.1.2.2 Outline

The outline should be set to the same color as the hatchings and will help to distinguish between runways of differing status more clearly. This is particularly noticeable when the view is not zoomed in such that a single runway fills the screen / viewing panel.

Each of the four images below show the same pair of runways. In each image, the eastern-most runway is closed. Note how much easier it is to distinguish the closed runway in the zoomed out view when the style includes an additional outline.

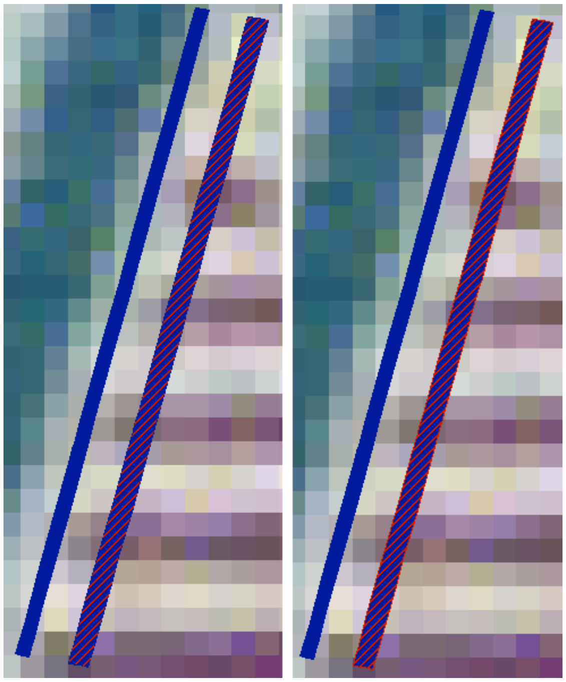


Figure 9 — Close-up of hatched filled Runway with (Right) and without (Left) additional outline

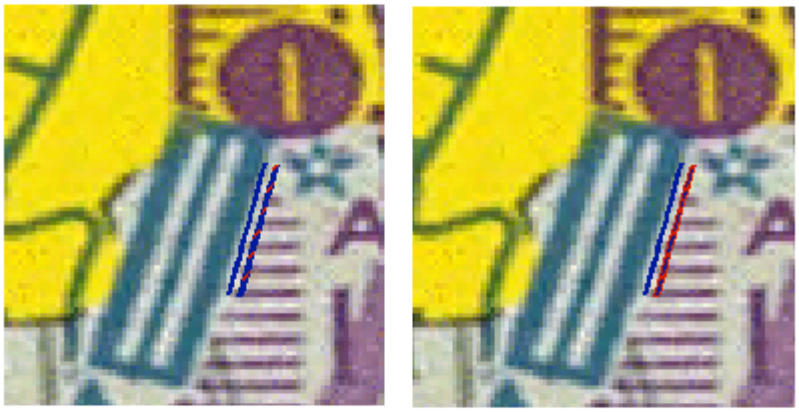


Figure 10 — Zoomed out view of hatched filled Runway with (Right) and without (Left) additional outline

7.3.1.3 Recommendations – Unserviceable features

Proposed symbols for indicating unserviceable navaids. Unservicable could be indicated by an overlay symbol consisting of a circle with a strike-through, angled 45 degrees referenced to the graphical display. To enhance contrast between symbol and overlay, the overlay below has a white outline, which also enhances readability in black and white or print for example. See figures below for suggested icons.

Table 1 - Unserviceable features

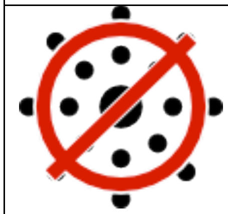
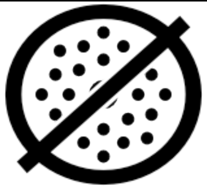
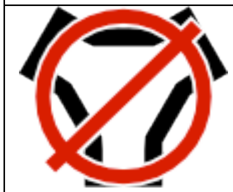
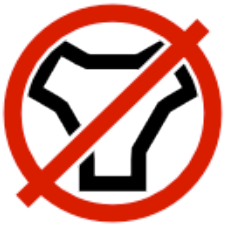
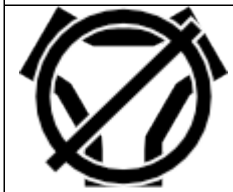
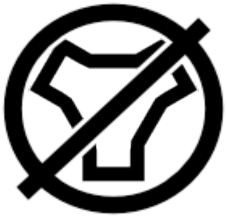
Icon	Comment
	Unservicable NDB with red overlay, smaller than the bounds of the original symbol.
	Unservicable NDB with black overlay, larger than the bounds of the original symbol. Makes for a rather large symbol unfortunately.
	Unservicable TACAN with red overlay, smaller than the bounds of the original symbol.
	Unservicable TACAN with red overlay, larger than the bounds of the original symbol.
	Unservicable TACAN with black overlay, smaller than the bounds of the original symbol.
	Unservicable TACAN with black overlay, larger than the bounds of the original symbol.

Figure 11 - Unserviceable navaids

7.3.2 Contamination

7.3.2.1 Problem

The ICAO Annex 4 contains no recommendation on how to portray contamination of surfaces, such as runways and taxiways.

7.3.2.2 Recommendations

In the digital SNOWTAM trial of Eurocontrol a triangular icon was used to indicate contaminations. The icon had a fill color based on friction estimate, and also showed the friction estimate together with the type of contaminant. Next to the icon, the extent of the upper contaminant layer was also visualized with a fill color based on contamination type. Se figures below.

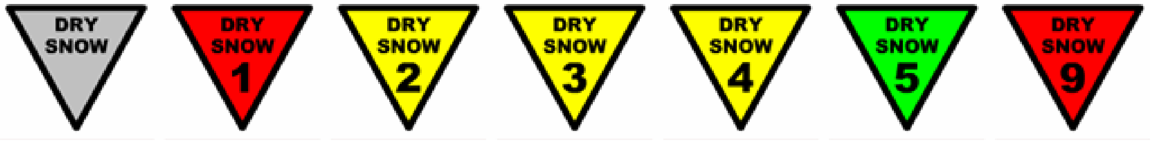


Figure 12 — SNOWTAM symbols.

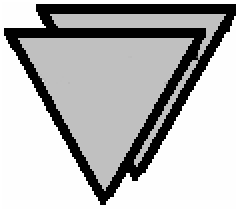


Figure 2 — Multiple layer contamination icon.

Contaminations are modeled as sub-objects of features, similar to light elements or markings. This means that we also have to use a hierarchical symbolizer.

7.3.2.2.1 Contamination data

Even if a surface has two or more layers with different horizontal projections, they should still be modelled as one SurfaceContamination, with several SurfaceContaminationLayer, but with different associated ElevatedSurface features. That's because there is only one observationTime. I this case a "multiple layer" symbol should be placed in the geometrical centre of the uppermost layer. The other ones should not be very different.

There is ongoing work on guidelines for runway friction measurement and reporting. This will most likely affect the SurfaceContamination model further.

7.3.2.2.2 Contamination Icon

The SE 1.2 SWG defines an update to how graphics can be defined in SLD. This can be used for the contamination icon. Note that this icon is supposed to be wrapped in a hierarchical symbolizer, so the xpaths are relative to an AbstractSurfaceContamination object.

The following is the outline of the SE specification for an icon, some blanks still need to be worked out:

```
<CompositeGraphic>
       <PointTextGraphic>
              <PointPosition><X>...</X><Y>...</Y></PointPosition>

            <PointLabel>aixm:layer/aixm:SurfaceContaminationLayer/
            aixm:type</PointLabel>
       </PointTextGraphic>
       <PointTextGraphic>
              <PointPosition><X>...</X><Y>...</Y></PointPosition>
              <PointLabel>aixm:frictionEstimation</PointLabel><!-
            -convert estimate to number-->
       </PointTextGraphic>
       <MarkGraphic>
              <WellKnownName>TRIANGLE</WellKnowName>
              <Fill>convert friction coefficient/estimate to
            color</Fill>
              <Stroke>thick black line</stroke>
       </MarkGraphic>
</CompositeGraphic>
```

7.3.2.2.3 Contamination Icon in SE 1.1.0

How can the contamination symbol be visualized in SE 1.1.0? This is a skeleton of a possible solution.

```
<!-- Contamination main symbol visualization -->
  <se:PointSymbolizer>
    <se:Graphic>
      <se:Mark>
        <se:WellKnownName>triangle</se:WellKnownName>
        <se:Fill>
          <se:SvgParameter name="fill">
            <!-- Recode into a color -->
            <se:Recode fallbackValue="UNDEFINED">
              <LookupValue>

      <ogc:PropertyName>aixm:frictionEstimation</ogc:PropertyName
      >
              </LookupValue>
              ...
            </se:SvgParameter>
        </se:Fill>
      </se:Mark>
      <se:Size>8.0</se:Size>
      <se:Rotation>180</se:Rotation>
    </se:Graphic>
  </se:PointSymbolizer>
  <se:TextSymbolizer>
    <se:Label>
      <!-- Recode into a number. Is that possible here? -->
      <se:Recode fallbackValue="UNDEFINED">
        <LookupValue>

     <ogc:PropertyName>aixm:frictionEstimation</ogc:PropertyName
     >
        </LookupValue>
        ...
      </se:Label>
  </se:TextSymbolizer>
  <se:TextSymbolizer>
    <se:Label>
     aixm:layer/aixm:SurfaceContaminationLayer/aixm:type<ogc:Pro
     pertyName></ogc:PropertyName>
    </se:Label>
  </se:TextSymbolizer>
```

7.4 Enterprise Considerations

As with any system that migrates to a computerized environment, it is likely that new requirements and use cases will be formed, driven by the flexibility and power offered by automated systems.

7.4.1 Algorithmic Rules

7.4.1.1 Problem

As the symbology business rules evolve, they are likely to include conditions based on more abstract queries of other feature types. For example, there may come a need to render a feature (let's say a runway) based on whether it intersects with the geometry of another feature (let's say an airspace that is closed due to volcanic ash). SLD does not allow the encoding of queries / conditions of this nature.

7.4.1.2 Option 1

Either the SE spec may need to consider extension in this area, or the wider system will need to place the responsibility for the necessary filtering query elsewhere. The pragmatic way forward could depend on the complexity of the required queries. In any case, some recommendations or guidelines for the AIM community might be desirable.

For example a portal viewing an area of interest might need to create the required application view by first firing off a WFS request to retrieve airspaces that are closed (or meet some other criteria that means runways, or other features within them are effected), and then fire off separate queries to retrieve affected and unaffected runways (by filtering based on the geometry retrieved from “affecting” airspaces). The portal would then ask the FPS to render a layer with affected runways with a very simple SLD, and then another layer with unaffected runways, again with a very simple SLD. In this way the portal (or more likely one of its application servers) would then own the domain specific business rules that cannot currently be encoded in an SLD.

One potential disadvantage of this approach is that geometries would need to be extracted by a portal component and embedded into SLDs for an FPS request. These geometries can be large and cumbersome to pass back and forth for such a transaction.

Of course allowing the SLD to specify rules / conditions of this nature would be a powerful extension to the standard and provide strong support to such a system.

7.4.1.3 Option 2 - Recommended

The suggested solution is an extension to the Rule element that allows multiple feature types to be specified. This would be semantically similar to the CSW specification. The existing Filter Encoding 2.0 specification would not need to change however the SE specification would need to define the semantic usage of the filter in this circumstance.

The example below is an extract from 07-110r4 CSW-ebRIM Registry Service (section 10.2.3) shows a similar requirement being addressed by the CSW specification.

```
<csw:Query typeNames="wrs:ExtrinsicObject rim:Association_a1">
 <csw:ElementSetName typeNames="wrs:ExtrinsicObject">full</csw:ElementSetName>
 <csw:Constraint version="1.1.0">
  <ogc:Filter>
   <ogc:And>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>$a1/@associationType</ogc:PropertyName>
     <ogc:Literal>urn:ogc:def:ebRIM-AssociationType:OGC:OperatesOn</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>$a1/@sourceObject</ogc:PropertyName>
     <ogc:Literal>urn:uuid:86084c6a-292f-4687-bf52-aef49b5ff2d6</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>wrs:ExtrinsicObject/@id</ogc:PropertyName>
     <ogc:PropertyName>$a1/@targetObject</ogc:PropertyName>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>wrs:ExtrinsicObject/@objectType</ogc:PropertyName>
     <ogc:Literal>urn:ogc:def:ebRIM-ObjectType:OGC:Dataset</ogc:Literal>
    </ogc:PropertyIsEqualTo>
   </ogc:And>
  </ogc:Filter>
 </csw:Constraint>
</csw:Query>
```

One possible encoding in the SLD would see the multiple type names allowed by CSW’s ElementSetName being reflected in SLD’s FeatureTypeStyle with a new element

"FeatureTypeNames". By applying the same semantics as the CSW example above, we would get something like this:

```
 <se:FeatureTypeStyle>
  <se:FeatureTypeNames>aixm:RunwayElement aixm:Airspace<se:FeatureTypeNames>
  <se:Rule>
   <ogc:Filter>
    <ogc:And>
     <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>RunwayElement/Category</ogc:PropertyName>
      <ogc:Literal>Paved</ogc:Literal >
     </ogc:PropertyIsEqualTo>
     <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>Airspace/Restriction</ogc:PropertyName>
      <ogc:Literal>Forbidden</ogc:Literal >
     </ogc:PropertyIsEqualTo>
     <fes:IsWithin>
      <!--spatial operand 1-->
      <fes:ValueReference>RunwayElement/Geometry</fes:ValueReference>
      <!--spatial operand 2-->
      <fes:ValueReference>Airspace/Geometry</fes:ValueReference>
     </fes:IsWithin>
    <ogc:And>
   </ogc:Filter>
   ... symbolizers ...
 </se:Rule>
</se:FeatureTypeStyle>
```

The semantic meaning of the above is to apply the symbolizers to "all paved runways within forbidden airspaces".

7.4.2 Delineations

Another potential evolution in portrayal requirements is that of "level of detail". Of course, with a digitally displayed chart, the ease at which the user can change the zoom / scale calls for well defined portrayal behaviors for doing so. These behaviors are likely to be specific to the function of the particular type of digital chart. However, they are also likely to involve the issue of the applicability of features' geometries at different scales.

This brings us to delineation of the geometry. When a user is zoomed in to a close view of runways and airports, they may reasonably expect to see the detailed polygonal geometry of the features. However, when zoomed out to a much larger "overview" scale covering a vast area, (for example North America) they may still wish to see the locations of runways or airports. At this scale though, it would probably be more appropriate to display the same features using point symbols rather than polygonal geometry. The polygonal geometry may well be invisible at this scale; at world scale the extent of a runway on a digital display could be less than a pixel (depending on the resolution of the display medium).

Level of detail could be supported by the WFS, which would then deliver the appropriate geometry for the requested scale. If this were the case, then the SLD may want to specify the conditions for its rules based on the delineation of the geometry. An example of an existing implementation of this extension to the SE specification is displayed below:

```
<Rule>
 <ogc:PropertyIsEqualTo>
  <ogc:Function name="geometryType">
   <ogc:PropertyName>geom</ogc:PropertyName>
  </ogc:Function>
  <ogc:Literal>Point</ogc:Literal>
 </ogc:PropertyIsEqualTo>
 <PointSymbolizer>
   ...
  </PointSymbolizer>
</Rule>
```

An extension to the SE specification should be considered in order to allow Rules to assess the type of geometry before applying symbolizers. The example above is clearly proprietary, and yet has cross-domain application. It would be a useful capability to add.

Alternatively, to address the simple case above, the SLD itself could be used to approximate the geometry using the polygons' centroid or centre of bounding box (this choice would need to be controllable from within the SLD). The scales could then be encoded in the SLD using existing constructs (MinScaleDenominator / MaxScaleDenominator) to control the level of detail trigger ranges.

7.4.2.1 Centre of Line and Polygon geometries - recommendation

When a PointSymbolizer is used with a Polygon or Line geometry, the specification says:

"In this case, if a line, polygon, or raster geometry is used with this Symbolizer, then the semantic is to use the centroid of the geometry, or any similar representative point."

It is recommended that a simple extension is made to the PointSymbolizer to disambiguate its usage under these circumstances. Furthermore, the choice of "centre-of" algorithm should also be a controllable aspect of the SLD. It should allow for centroid, centre of bounding box, and for polygons a flag to force the resultant point to fall inside the polygon (consider a moon crescent shaped polygon)

TBD – example of formal encoding of these options. Consider - are there any further common centre-of algorithms to include?

Annex A - Aviation Portrayal based on Directional Associations in AIXM

1 Introduction

1.1 Purpose and Scope

This technical note has been produced by Envitia to summarise some of the issues encountered when applying the OGC Symbology Encoding (SE) to the area of aviation portrayal. In particular, it concentrates on the issue of portrayal rules that are dependent on associations between feature types in the AIXM conceptual model, but which are not realised explicitly via GML properties within the AIXM GML application schema.

This report is the result of ongoing research by Envitia within the OGC OWS-8 Aviation Thread [6]. The issue with AIXM associations can be traced to specific requirements in the ICAO portrayal specifications that were studied in OWS-8 [4].

Section 2 describes the problem space in more detail, while Section 3 proposes some solutions.

1.2 Abbreviations

AIXM	Aeronautical Information Exchange Model
Envitia	Envitia Ltd
EUROCONTROL	European Organisation for the Safety of Air Navigation
FAA	Federal Aviation Administration
FPS	Feature Portrayal Service
GML	Geography Markup Language
ICAO	International Civil Aviation Organization
ISO	International Organization for Standardization
ISO/TC	ISO / Technical Committee
OGC	Open Geospatial Consortium
OWS	OGC Web Services
SE	Symbology Encoding
SLD	Styled Layer Descriptor

UML	Unified Modelling Language
XML	Extensible Markup Language
XSD	XML Schema

1.3 References and Supporting documents

The following documents are referenced by, or are directly relevant to, this document:

1. AIXM Conceptual Model, version 4.5, EUROCONTROL, 2007
2. AIXM Conceptual Model, version 5.1, FAA, EUROCONTROL, 2010
3. AIXM XML Schema, version 5.1, FAA, EUROCONTROL, 2010
4. Convention on International Civil Aviation, Annex 4 – Aeronautical Charts, 11th Edition, ICAO, 2009
5. ISO 19136:2007, Geographic information – Geography Markup Language (GML).
6. OGC OWS-8 Engineering Report - Guidelines for ICAO portrayal using SLD/SE, OGC, in preparation, 2011
7. OGC 05-077r4, Symbology Encoding Implementation Specification, v.1.1.0, 2006
8. OGC 05-078r4, Styled Layer Descriptor profile of the Web Map Service Implementation Specification, v.1.1.0, 2007

2 Directional Associations in the AIXM Application Schema and Ramifications to Portrayal

One of the topics of the OGC OWS-8 Aviation Thread is aviation styling and portrayal tasks (including ICAO guidance for SLD) and the development of Feature Portrayal Services (FPS). It was established from the outset that the OWS-8 research should concentrate on some of the more challenging aspects of accommodating ICAO symbology in SLD/SE, and not simply address the ‘easy’ cases. As such, a number of portrayal rules have been identified where the symbology applied to a given feature type, such as Aerodrome, is dependent on associated feature types, such as Runway. This situation is not peculiar to the aeronautical domain. For example, in the hydrographic domain, an analogous example is where the symbology used to portrayal a Buoy or Beacon is dependent on associated components such as its Light and Fog Signal.

To highlight a particular ICAO portrayal rule, refer to Appendix 2 – ICAO Chart Symbols – of [4]. According to APP 2-17, the symbology for an Aerodrome may be augmented with abbreviated textual data such as the Aerodrome’s elevation, length of its longest Runway and the Runway surface type.

The focus of this technical note is the dependency of the Aerodrome symbology on the associated Runway feature, and how this may be handled using the OGC Symbology Encoding specification.[1] The current SE specification supports this type of portrayal rule, where the symbol to be rendered is dependent of the value of a property of the feature type. However, the SE specification requires the portrayal condition to be an OGC Filter expression in terms of *properties of the GML feature type* that is being portrayed. In the case of associated feature types, such as Aerodrome and Runway, the condition consists of XPath expressions that navigate from the source feature type, along a complex-valued GML property, to the target feature type. As an example from AIXM, the following SE fragment symbolises an AIXM AirportHeliport feature based on its aircraft type. Note that the XPath expression navigates from the source AirportHeliport[2] feature type, through associated feature types such as AirportHeliportAvailability and AirportHeliportUsage, to reach the ‘aircraft type’ property that is used to distinguish the resulting symbol choice.

```
<sld:UserStyle>
  <se:FeatureTypeStyle>
    <se:Name>Airport Heliport</se:Name>
    <se:FeatureTypeName>aixm:AirportHeliport</se:FeatureTypeName>
    <se:Rule>
      <ogc:Filter>
```

[1] The Aerodrome symbology described in APP 2-17 introduces other complications, not least the possibility of an Aerodrome having multiple Runways and the need to base the symbology on the Runway having longest length. Such considerations are not the focus of this technical paper.

[2] Technically, the first property targets the AirportHeliportTimeSlice element that is a component of the AIXM temporal model. The AirportHeliportPropertyGroup defines the XML content model representing the AirportHeliport properties per-se; see Figure 15.

```
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
aixm:timeSlice/aixm:AirportHeliportTimeSlice/aixm:availability/aixm:Airport
HeliportAvailability/aixm:usage/aixm:AirportHeliportUsage/aixm:selection/ai
xm:ConditionCombination/aixm:aircraft/aixm:AircraftCharacteristic/aixm:type
</ogc:PropertyName>
          <ogc:Literal>HELICOPTER</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <se:PointSymbolizer>
        <se:Geometry>
          <ogc:PropertyName>geometry</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:ExternalGraphic>
            <se:OnlineResource
xlink:href="http://services.envitia.com/images/ows-8/Heliport.png"
xlink:type="simple"/>
            <se:Format>image/png</se:Format>
          </se:ExternalGraphic>
          <se:Size>
            <ogc:Literal>30</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
    <se:Rule>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
aixm:timeSlice/aixm:AirportHeliportTimeSlice/aixm:availability/aixm:Airport
HeliportAvailability/aixm:usage/aixm:AirportHeliportUsage/aixm:selection/ai
xm:ConditionCombination/aixm:aircraft/aixm:AircraftCharacteristic/aixm:type
</ogc:PropertyName>
          <ogc:Literal>LANDPLANE</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <se:PointSymbolizer>
        <se:Geometry>
          <ogc:PropertyName>geometry</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:ExternalGraphic>
            <se:OnlineResource
xlink:href="http://services.envitia.com/images/ows-8/CivilAerodrome.png"
xlink:type="simple"/>
            <se:Format>image/png</se:Format>
          </se:ExternalGraphic>
          <se:Size>
            <ogc:Literal>30</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
    <!-- ... -->
  </se:FeatureTypeStyle>
</sld:UserStyle>
```

Figure 13 - SE Fragment Symbolizing Aerodrome based on its Aircraft Type

The above SE encoding is valid because the AIXM GML application schema provides properties for navigating along the indicated XPath expressions. However, the same is not true of the AirportHeliport – Runway association. In UML terms, the AirportHeliport – Runway association is uni-directional so, according to the UML to GML encoding rules, only the navigable association role is realised as a GML property in the GML application schema.[3] This is shown by the following AIXM UML application schema fragment.

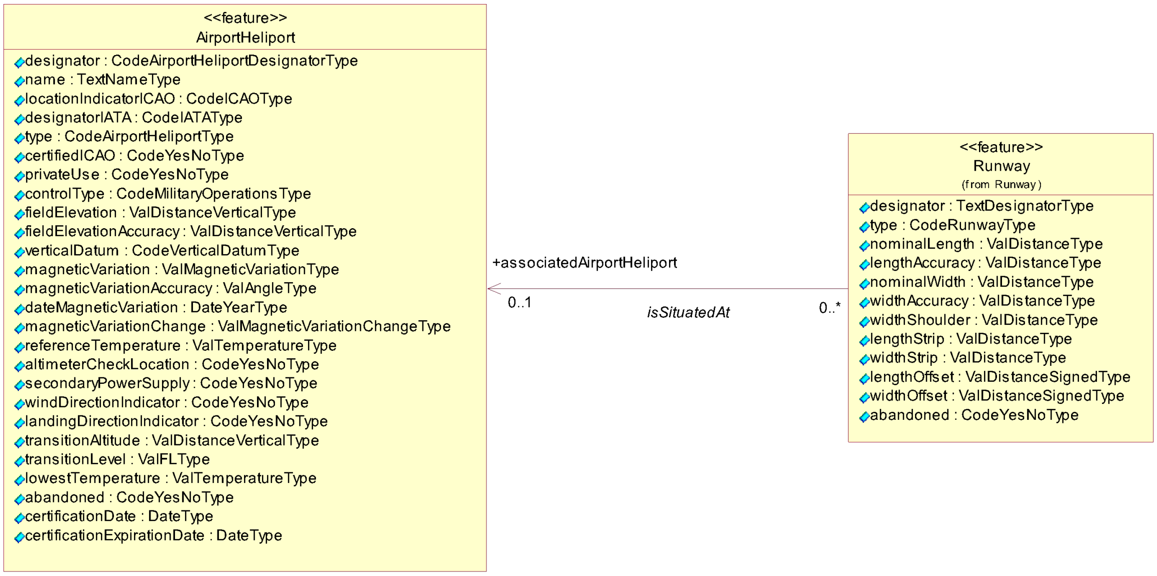

<<feature>>
AirportHeliport
designator : CodeAirportHeliportDesignatorType
name : TextNameType
locationIndicatorICAO : CodeICAOType
designatorIATA : CodeIATAType
type : CodeAirportHeliportType
certifiedICAO : CodeYesNoType
privateUse : CodeYesNoType
controlType : CodeMilitaryOperationsType
fieldElevation : ValDistanceVerticalType
fieldElevationAccuracy : ValDistanceVerticalType
verticalDatum : CodeVerticalDatumType
magneticVariation : ValMagneticVariationType
magneticVariationAccuracy : ValAngleType
dateMagneticVariation : DateYearType
magneticVariationChange : ValMagneticVariationChangeType
referenceTemperature : ValTemperatureType
altimeterCheckLocation : CodeYesNoType
secondaryPowerSupply : CodeYesNoType
windDirectionIndicator : CodeYesNoType
landingDirectionIndicator : CodeYesNoType
transitionAltitude : ValDistanceVerticalType
transitionLevel : ValFLType
lowestTemperature : ValTemperatureType
abandoned : CodeYesNoType
certificationDate : DateType
certificationExpirationDate : DateType
+associatedAirportHeliport
0..1
isSituatedAt
0..*
<<feature>>
Runway
(from Runway)
designator : TextDesignatorType
type : CodeRunwayType
nominalLength : ValDistanceType
lengthAccuracy : ValDistanceType
nominalWidth : ValDistanceType
widthAccuracy : ValDistanceType
widthShoulder : ValDistanceType
lengthStrip : ValDistanceType
widthStrip : ValDistanceType
lengthOffset : ValDistanceSignedType
widthOffset : ValDistanceSignedType
abandoned : CodeYesNoType

Figure 14 - AIXM 5.1 UML – AirportHeliport / Runway Association

The AIXM GML application schema is derived from the UML application schema. Accordingly, the GML application schema representation of AirportHeliport and Runway is as follows. It should be noted that there is no GML property allowing navigation from the AirportHeliport to its associated Runway, but it is possible to navigate in the reverse direction via the associatedAirportHeliport property.

```
<group name="AirportHeliportPropertyGroup">
  <sequence>
    <element name="designator" type="aixm:CodeAirportHeliportDesignatorType"
nillable="true" minOccurs="0"/>
    <element name="name" type="aixm:TextNameType" nillable="true" minOccurs="0"/>
    <element name="locationIndicatorICAO" type="aixm:CodeICAOType" nillable="true"
minOccurs="0"/>
    <element name="designatorIATA" type="aixm:CodeIATAType" nillable="true"
minOccurs="0"/>
    <element name="type" type="aixm:CodeAirportHeliportType" nillable="true"
minOccurs="0"/>
```

[3] In UML terms, Associations are first-class entities in the UML meta-model and exist independently of the realisation of any association 'roles' as properties bound to the feature types at the association ends. In contrast, in GML terms (and also in object-oriented programming languages such as C++ and Java) the association is typically implemented in terms of its association roles.

```
    <element name="certifiedICAO" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="privateUse" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="controlType" type="aixm:CodeMilitaryOperationsType"
nillable="true" minOccurs="0"/>
    <element name="fieldElevation" type="aixm:ValDistanceVerticalType"
nillable="true" minOccurs="0"/>
    <element name="fieldElevationAccuracy" type="aixm:ValDistanceVerticalType"
nillable="true" minOccurs="0"/>
    <element name="verticalDatum" type="aixm:CodeVerticalDatumType" nillable="true"
minOccurs="0"/>
    <element name="magneticVariation" type="aixm:ValMagneticVariationType"
nillable="true" minOccurs="0"/>
    <element name="magneticVariationAccuracy" type="aixm:ValAngleType"
nillable="true" minOccurs="0"/>
    <element name="dateMagneticVariation" type="aixm:DateYearType" nillable="true"
minOccurs="0"/>
    <element name="magneticVariationChange"
type="aixm:ValMagneticVariationChangeType" nillable="true" minOccurs="0"/>
    <element name="referenceTemperature" type="aixm:ValTemperatureType"
nillable="true" minOccurs="0"/>
    <element name="altimeterCheckLocation" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="secondaryPowerSupply" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="windDirectionIndicator" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="landingDirectionIndicator" type="aixm:CodeYesNoType"
nillable="true" minOccurs="0"/>
    <element name="transitionAltitude" type="aixm:ValDistanceVerticalType"
nillable="true" minOccurs="0"/>
    <element name="transitionLevel" type="aixm:ValFLType" nillable="true"
minOccurs="0"/>
    <element name="lowestTemperature" type="aixm:ValTemperatureType" nillable="true"
minOccurs="0"/>
    <element name="abandoned" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="certificationDate" type="aixm:DateType" nillable="true"
minOccurs="0"/>
    <element name="certificationExpirationDate" type="aixm:DateType" nillable="true"
minOccurs="0"/>
    <element name="contaminant" type="aixm:AirportHeliportContaminationPropertyType"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    <element name="servedCity" type="aixm:CityPropertyType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
    <element name="responsibleOrganisation"
type="aixm:AirportHeliportResponsibilityOrganisationPropertyType" nillable="true"
minOccurs="0"/>
    <element name="ARP" type="aixm:ElevatedPointPropertyType" nillable="true"
minOccurs="0"/>
    <element name="aviationBoundary" type="aixm:ElevatedSurfacePropertyType"
nillable="true" minOccurs="0"/>
    <element name="altimeterSource" type="aixm:AltimeterSourcePropertyType"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    <element name="contact" type="aixm:ContactInformationPropertyType"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    <element name="availability" type="aixm:AirportHeliportAvailabilityPropertyType"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    <element name="annotation" type="aixm:NotePropertyType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
```

```
  </sequence>
</group>
```

Figure 15 - AIXM 5.1 XSD – AirportHeliport Properties

```
<group name="RunwayPropertyGroup">
  <sequence>
    <element name="designator" type="aixm:TextDesignatorType"
nillable="true" minOccurs="0"/>
    <element name="type" type="aixm:CodeRunwayType" nillable="true"
minOccurs="0"/>
    <element name="nominalLength" type="aixm:ValDistanceType"
nillable="true" minOccurs="0"/>
    <element name="lengthAccuracy" type="aixm:ValDistanceType"
nillable="true" minOccurs="0"/>
    <element name="nominalWidth" type="aixm:ValDistanceType"
nillable="true" minOccurs="0"/>
    <element name="widthAccuracy" type="aixm:ValDistanceType"
nillable="true" minOccurs="0"/>
    <element name="widthShoulder" type="aixm:ValDistanceType"
nillable="true" minOccurs="0"/>
    <element name="lengthStrip" type="aixm:ValDistanceType" nillable="true"
minOccurs="0"/>
    <element name="widthStrip" type="aixm:ValDistanceType" nillable="true"
minOccurs="0"/>
    <element name="lengthOffset" type="aixm:ValDistanceSignedType"
nillable="true" minOccurs="0"/>
    <element name="widthOffset" type="aixm:ValDistanceSignedType"
nillable="true" minOccurs="0"/>
    <element name="abandoned" type="aixm:CodeYesNoType" nillable="true"
minOccurs="0"/>
    <element name="surfaceProperties"
type="aixm:SurfaceCharacteristicsPropertyType" nillable="true"
minOccurs="0"/>
    <element name="associatedAirportHeliport"
type="aixm:AirportHeliportPropertyType" nillable="true" minOccurs="0"/>
    <element name="overallContaminant"
type="aixm:RunwayContaminationPropertyType" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
    <element name="annotation" type="aixm:NotePropertyType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
    <element name="areaContaminant"
type="aixm:RunwaySectionContaminationPropertyType" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</group>
```

Figure 16 - AIXM 5.1 XSD – Runway Properties

It is therefore not possible to express the ICAO portrayal rules for AirportHeliport, which place a dependency on its related Runway, using the current AIXM application schema and OGC SE specification.

3 AIXM Portrayal using OGC Symbology Encoding

Possible solutions to the problem raised in Section 2 are proposed in this Section.

3.1 Revise the AIXM Application Schema

There is a strong case for revising the AIXM UML application schema to use bi-directional associations, from a pragmatic and technical perspective, but also from a conceptual perspective.

The use of uni-directional UML associations, and the resulting lack of 'reverse-properties' at the GML level, imposes technical limitations on any client application that needs to efficiently navigate between the associated features, not just portrayal-based clients. In the case of the AirportHeliport – Runway association there appears to no domain-specific reason for this constraint. One possible explanation comes from the AIXM 4.5 Conceptual Model. This predecessor to AIXM 5 used an entity-relationship modelling paradigm and relationships were named, and directed, based on the underlying relational model. In the case of AirportHeliport – Runway the relationship goes from Runway to AirportHeliport.

On moving from AIXM 4.5 to AIXM 5, the ISO/TC 211 approach was adopted and the AIXM conceptual model was represented using UML. The directionality of the associations from AIXM 4.5 appears to have been translated to association role navigability – this becomes problematic. At the conceptual model level a bi-directional association appears to be more appropriate. The directionality of the association name may still be maintained in AIXM 5 using the UML triangle notation on the association label, as shown in the revised UML model below.

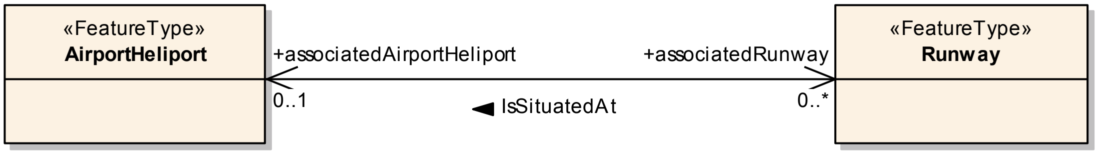


Figure 17 - Revised AIXM UML – AirportHeliport / Runway Association

The AIXM GML application schema is then derived from the UML application schema and the resulting AirportHeliport feature type would now contain a complex-valued property associatedRunway, linking it explicitly with its Runway feature(s). SE portrayal rules for AirportHeliport may now be formulated as XPath expressions in terms of the associatedRunway property.

Note: an 'interim' solution along these lines has been explored in the OWS-8 thread whereby existing AIXM 5.1 GML feature types were extended at the XML Schema level to include 'reverse-properties' such as associatedRunway.

3.2 Extend the OGC Symbology Encoding

Revision of the AIXM application schema represents a significant re-issue of AIXM.

From a portrayal sevice perspective, if the service is required to portray AIXM GML instance data from some external WFS then it is highly desirable for the GML

AirportHeliport feature type to support an associatedRunway property linking it explicitly to its Runway feature(s). However, if this property is not available it would still be possible for the service to portray the AirportHeliport provided it could determine the related Runway instances by some alternate mechanism. Indeed, if the service was not a Feature Portrayal Service in the generic sense, but a service offering portrayal of AIXM instance data that is stored natively in an internal relational database, then the service could efficiently determine the Runway(s) related to a given AirportHeliport via a straightforward relational join query.

The SE needs to be extended to allow the specification of 'Runway associated with given AirportHeliport' in the absence of an explicit associatedRunway property on the AirportHeliport feature type. The proposed encoding is shown below.

```
<env:AssociatedProperty>
  <env:sourceRoleName>aixm:associatedAirportHeliport</env:sourceRoleName>
  <env:targetPropertName>
aixm:Runway/aixm:timeSlice/aixm:RunwayTimeSlicePropertyType/nominalLength</
env:targetPropertName>
</env:AssociatedProperty>
```

Figure 18 - Referring to Runway nominalLength via AssociatedProperty

The new SE element AssociatedProperty identifies the property on the remote feature type (in this case Runway) to be used in the portrayal condition. This is a complex type with the following properties:

- sourceRoleName: name of the association role at the source (AirportHeliport) end of the association.
- associationName: name of the association.
- targetPropertName: name of the property at the target (Runway) end of the association. This is an XPath expression that includes the value-type at the target end, allowing for dynamic substitution of more specialised sub-types.

Note: sourceRoleName and associationName provide alternative means of identifying the link: one or the other should be supplied. A simplified XSD representation of the AssociatedProperty element (using xsd:string) is given below.

```
<xsd:element name="AssociatedProperty" type="env:AssociatedPropertyType">
  <xsd:annotation>
    <xsd:documentation>An AssociatedProperty identifies the property on the
remote feature type, linked to the source feature type via association. The
association may be identified by its name or the name of the association
role at the source end of the association.</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="AssociatedPropertyType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="sourceRoleName" type="xsd:string"/>
      <xsd:element name="associationName" type="xsd:string"/>
    </xsd:choice>
    <xsd:element name="targetPropertyName" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

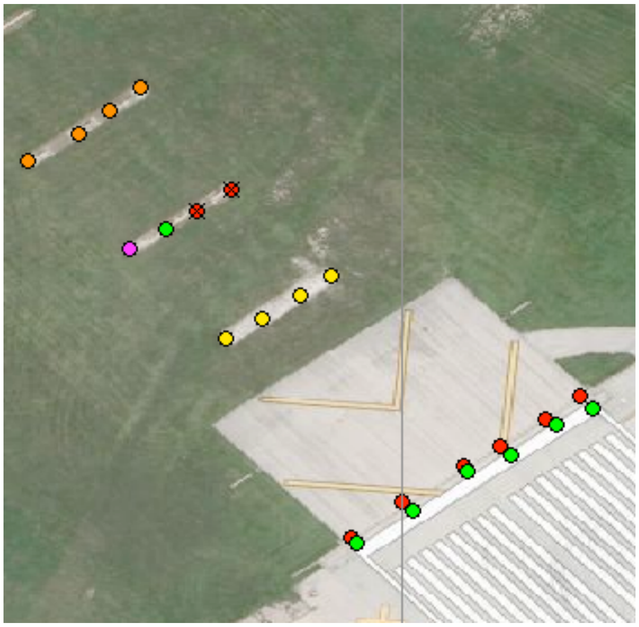
Figure 19 - XSD Representation of AssociatedProperty

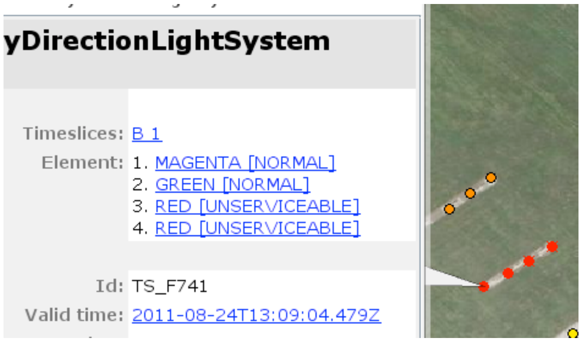
Annex B - Styling of AIXM 5.1 light systems using composite rules

1 Introduction

A new type of composite rule has been defined to enable styling of nested objects. Nested objects occur very often in both AIXM 5.1 and WXXM 1.1, which should illustrate the relevance of the new type of rule.

This is an example of a light system that is styled with this new type of rule:





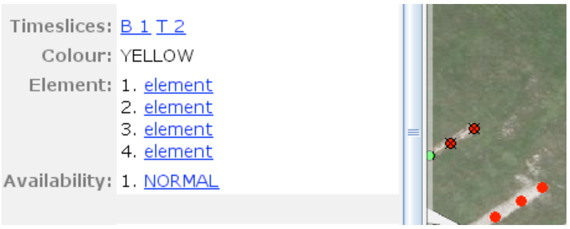
yDirectionLightSystem

Timeslices: B 1

Element: 1. MAGENTA [NORMAL]
2. GREEN [NORMAL]
3. RED [UNSERVICEABLE]
4. RED [UNSERVICEABLE]

Id: TS_F741

Valid time: 2011-08-24T13:09:04.479Z



Timeslices: B 1 T 2

Colour: YELLOW

Element: 1. element
2. element
3. element
4. element

Availability: 1. NORMAL

Notice how one light system has elements with colors and availability defined in the separate elements, while the other one has a global color and availability. This complexity is part of the motivation for requiring this new type of rule.

These are the formal reasons for the change:

1. 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.
2. Another CR defines a CompositeChildSymbolizer to apply a symbolizer to child objects, this is not always sufficient because there are examples where we need to apply rules to nested objects to select the right symbolizer.
3. Currently the concept of 'else if' as defined by many programming languages is not supported by SE. This makes rules hard to write and inefficient, as every rule also needs a condition that is the inverse of the conditions of the other rules to

achieve the same effect.

The following elements and attributes are added with respect to se:RuleType:

1. se:Rule (0..*) A list of rules
2. se:ChildProperty (0..1) A property pointing to a list of nested objects.
3. elseIfRule A boolean indicating that the rules contained in this composite rule should be treated as an 'Else If' block.

We define the following semantics:

- The CompositeRuleType extends from RuleType, this includes semantics.
- If a ChildProperty is set, all symbolizers and rules that are part of this rule are applied to each element of the list that is obtained by evaluating the ChildProperty against the object that was accepted by this rule.
- If elseIfRule is set to true, the rules are evaluated in the same order as they appear in the list, only the first rule that accepts a given (nested) object is used for styling. If elseIfRule is set to false, the semantics that are now used in an se:FeatureTypeStyle apply.
- The use of the XPath parent axis '..' is allowed to reference from a nested object back to one of its parents

1.1 Composite rule schema

The following custom extension to the SE schema was used:

```
<xsd:element name="CompositeRule" type="ows:CompositeRuleType"
substitutionGroup="se:Rule"/>
<xsd:complexType name="CompositeRuleType" >
  <xsd:complexContent>
    <xsd:extension base="se:RuleType">
      <xsd:sequence>
        <xsd:element ref="se:Rule" maxOccurs="unbounded"/>
        <xsd:element name="ChildProperty" type="ogc:PropertyNameType"/>
      </xsd:sequence>
      <xsd:attribute name="elseIfRule" type="xsd:boolean" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

1.2 SE example

This is a complete style that uses composite rules. It covers a few ways of how this new type of rule can be used.

```
<?xml version='1.0' encoding='UTF-8'?>
<se:FeatureTypeStyle xmlns:se="http://www.opengis.net/se"
xmlns:ows8="http://www.opengis.net/se/ows8"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:aixm="http://www.aixm.aero/schema/5.1" version="1.1.0">
  <se:Name>RunwayDirectionLightSystemType</se:Name>
  <se:FeatureTypeName>RunwayDirectionLightSystemType</se:FeatureTypeName>
  <se:Rule>
    <se:ElseFilter/>
    <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
    <se:MaxScaleDenominator>25000.0</se:MaxScaleDenominator>
    <se:PointSymbolizer>
      <se:Geometry>
<ogc:PropertyName>aixm:timeSlice/aixm:RunwayDirectionLightSystemTimeSlice/a
ixm:element/aixm:LightElement/aixm:location/aixm:ElevatedPoint</ogc:Propert
yName>
      </se:Geometry>
      <se:Graphic>
        <se:Mark>
          <se:WellKnownName>circle</se:WellKnownName>
          <se:Fill>
            <se:SvgParameter name="fill">#808080</se:SvgParameter>
          </se:Fill>
          <se:Stroke>
            <se:SvgParameter name="stroke">#000000</se:SvgParameter>
          </se:Stroke>
        </se:Mark>
        <se:Size>
          <ogc:Literal>8</ogc:Literal>
        </se:Size>
      </se:Graphic>
    </se:PointSymbolizer>
  </se:Rule>
  <ows8:CompositeRule  elseIfRule="true">
    <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
    <se:MaxScaleDenominator>25000.0</se:MaxScaleDenominator>
    <se:Rule>
      <ogc:Filter>
        <ogc:Not>
          <ogc:PropertyIsNull>
            <ogc:PropertyName>aixm:colour</ogc:PropertyName>
          </ogc:PropertyIsNull>
        </ogc:Not>
      </ogc:Filter>
      <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
      <se:MaxScaleDenominator>INF</se:MaxScaleDenominator>
      <se:PointSymbolizer>
        <se:Geometry>
<ogc:PropertyName>aixm:location/aixm:ElevatedPoint</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:Mark>
            <se:WellKnownName>circle</se:WellKnownName>
            <se:Fill>
              <se:SvgParameter name="fill">
                <ogc:PropertyName>aixm:colour</ogc:PropertyName>
              </se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
            </se:Stroke>
          </se:Mark>
          <se:Size>
            <ogc:Literal>8</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
    <se:Rule>
      <ogc:Filter>
        <ogc:Not>
          <ogc:PropertyIsNull>
            <ogc:PropertyName>../../aixm:colour</ogc:PropertyName>
          </ogc:PropertyIsNull>
        </ogc:Not>
      </ogc:Filter>
      <se:PointSymbolizer>
        <se:Geometry>
<ogc:PropertyName>aixm:location/aixm:ElevatedPoint</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:Mark>
            <se:WellKnownName>circle</se:WellKnownName>
            <se:Fill>
              <se:SvgParameter name="fill">
                <ogc:PropertyName>../../aixm:colour</ogc:PropertyName>
              </se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
            </se:Stroke>
          </se:Mark>
          <se:Size>
            <ogc:Literal>8</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
    <se:Rule>
      <se:PointSymbolizer>
        <se:Geometry>
<ogc:PropertyName>aixm:location/aixm:ElevatedPoint</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:Mark>
            <se:WellKnownName>circle</se:WellKnownName>
            <se:Fill>
              <se:SvgParameter name="fill">#808080</se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
            </se:Stroke>
          </se:Mark>
          <se:Size>
            <ogc:Literal>8</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
<ows8:ChildProperty>aixm:timeSlice/aixm:RunwayDirectionLightSystemTimeSlice
/aixm:element/aixm:LightElement</ows8:ChildProperty>
  </ows8:CompositeRule>
  <ows8:CompositeRule>
    <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
    <se:MaxScaleDenominator>25000.0</se:MaxScaleDenominator>
    <se:Rule>
      <ogc:Filter>
        <ogc:Or>
          <ogc:And>
            <ogc:Not>
              <ogc:PropertyIsEqualTo matchCase="true">
                <ogc:PropertyName
>aixm:availability/aixm:LightElementStatus/aixm:status</ogc:PropertyName>
                <ogc:Literal>NORMAL</ogc:Literal>
              </ogc:PropertyIsEqualTo>
            </ogc:Not>
            <ogc:Not>
              <ogc:PropertyIsNull>
<ogc:PropertyName>aixm:availability/aixm:LightElementStatus/aixm:status</og
c:PropertyName>
              </ogc:PropertyIsNull>
            </ogc:Not>
          </ogc:And>
          <ogc:And>
            <ogc:Not>
              <ogc:PropertyIsNull>
<ogc:PropertyName>../../aixm:availability/aixm:GroundLightingAvailability/a
ixm:operationalStatus</ogc:PropertyName>
              </ogc:PropertyIsNull>
            </ogc:Not>
            <ogc:Not>
              <ogc:PropertyIsEqualTo matchCase="true">
<ogc:PropertyName>../../aixm:availability/aixm:GroundLightingAvailability/a
ixm:operationalStatus</ogc:PropertyName>
                <ogc:Literal>NORMAL</ogc:Literal>
              </ogc:PropertyIsEqualTo>
            </ogc:Not>
          </ogc:And>
        </ogc:Or>
      </ogc:Filter>
      <se:PointSymbolizer>
        <se:Geometry>
<ogc:PropertyName>aixm:location/aixm:ElevatedPoint</ogc:PropertyName>
        </se:Geometry>
        <se:Graphic>
          <se:Mark>
            <se:WellKnownName>x</se:WellKnownName>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
            </se:Stroke>
          </se:Mark>
          <se:Size>
            <ogc:Literal>8</ogc:Literal>
          </se:Size>
        </se:Graphic>
      </se:PointSymbolizer>
    </se:Rule>
<ows8:ChildProperty>aixm:timeSlice/aixm:RunwayDirectionLightSystemTimeSlice
/aixm:element/aixm:LightElement</ows8:ChildProperty>
  </ows8:CompositeRule>
</se:FeatureTypeStyle>
```

1.3 Data sample

This is an example of a data set that can be used with the style given above. It covers most of the rules that are defined.

```
<?xml version='1.0' encoding='UTF-8'?>
<ns1:AIXMBasicMessage xmlns:ns1="http://www.aixm.aero/schema/5.1/message"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:xlink="http://www.w3.org/1999/xlink" gml:id="messageId">
  <ns1:hasMember>
    <aixm:RunwayDirectionLightSystem gml:id="F734">
      <gml:identifier codeSpace="urn:uuid">RDLS1</gml:identifier>
      <aixm:timeSlice>
        <aixm:RunwayDirectionLightSystemTimeSlice gml:id="TS735">
          <gml:validTime>
            <gml:TimePeriod gml:id="VT736">
              <gml:beginPosition>2010-10-
01T08:59:00.000Z</gml:beginPosition>
              <gml:endPosition
indeterminatePosition="unknown"></gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:timeSliceMetadata/>
          <aixm:colour>YELLOW</aixm:colour>
          <aixm:element>
            <aixm:LightElement gml:id="id737">
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P738">
                  <gml:pos>-87.93374893841418 41.990784733743546</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id737">
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P738">
                  <gml:pos>-87.93366026019484 41.99083141967231</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id737">
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P738">
                  <gml:pos>-87.93357410007476 41.99088680832093</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id737">
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P738">
                  <gml:pos>-87.93349819711186 41.99093604267526</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
            </aixm:LightElement>
          </aixm:element>
          <aixm:availability>
            <aixm:GroundLightingAvailability gml:id="id740">
              <aixm:operationalStatus>NORMAL</aixm:operationalStatus>
            </aixm:GroundLightingAvailability>
          </aixm:availability>
        </aixm:RunwayDirectionLightSystemTimeSlice>
      </aixm:timeSlice>
      <aixm:timeSlice>
        <aixm:RunwayDirectionLightSystemTimeSlice gml:id="TS735">
          <gml:validTime>
            <gml:TimePeriod gml:id="VT736">
              <gml:beginPosition>2010-12-
06T08:59:00.000Z</gml:beginPosition>
              <gml:endPosition>2011-03-06T08:59:00.000Z</gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>TEMPDELTA</aixm:interpretation>
          <aixm:sequenceNumber>2</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:availability>
            <aixm:GroundLightingAvailability gml:id="id740">

<aixm:operationalStatus>UNSERVICEABLE</aixm:operationalStatus>
            </aixm:GroundLightingAvailability>
          </aixm:availability>
        </aixm:RunwayDirectionLightSystemTimeSlice>
      </aixm:timeSlice>
    </aixm:RunwayDirectionLightSystem>
  </ns1:hasMember>
  <ns1:hasMember>
    <aixm:RunwayDirectionLightSystem gml:id="F741">
      <gml:identifier codeSpace="urn:uuid">RDLS2</gml:identifier>
      <aixm:timeSlice>
        <aixm:RunwayDirectionLightSystemTimeSlice gml:id="TS742">
          <gml:validTime>
            <gml:TimePeriod gml:id="VT743">
              <gml:beginPosition>2010-10-
01T08:59:00.000Z</gml:beginPosition>
              <gml:endPosition
indeterminatePosition="unknown"></gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:timeSliceMetadata/>
          <aixm:element>
            <aixm:LightElement gml:id="id744">
              <aixm:colour>MAGENTA</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P745">
                  <gml:pos>-87.93397412920366 41.99099963704959</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
                <aixm:LightElementStatus gml:id="id746">
                  <aixm:status>NORMAL</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id744">
              <aixm:colour>GREEN</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P745">
                  <gml:pos>-87.9338879690836 41.99104476854106</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id744">
              <aixm:colour>RED</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P745">
                  <gml:pos>-87.93381411755212 41.99108990003252</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
                <aixm:LightElementStatus gml:id="id746">
                  <aixm:status>UNSERVICEABLE</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
        </aixm:RunwayDirectionLightSystemTimeSlice>
      </aixm:timeSlice>
    </aixm:RunwayDirectionLightSystem>
  </ns1:hasMember>
  <ns1:hasMember>
    <aixm:RunwayDirectionLightSystem gml:id="F748">
      <gml:identifier codeSpace="urn:uuid">RDLS3</gml:identifier>
      <aixm:timeSlice>
        <aixm:RunwayDirectionLightSystemTimeSlice gml:id="TS749">
          <gml:validTime>
            <gml:TimePeriod gml:id="VT750">
              <gml:beginPosition>2010-10-
01T08:59:00.000Z</gml:beginPosition>
              <gml:endPosition
indeterminatePosition="unknown"></gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:timeSliceMetadata/>
          <aixm:element>
            <aixm:LightElement gml:id="id751">
              <aixm:colour>ORANGE</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P752">
                  <gml:pos>-87.93421242347861 41.99120783682546</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
```

```
                <aixm:LightElementStatus>
                  <aixm:status>NORMAL</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id751">
              <aixm:colour>ORANGE</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P752">
                  <gml:pos>-87.93409454822864 41.99127370828868</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
                <aixm:LightElementStatus>
                  <aixm:status>NORMAL</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
        </aixm:RunwayDirectionLightSystemTimeSlice>
      </aixm:timeSlice>
    </aixm:RunwayDirectionLightSystem>
  </ns1:hasMember>
  <ns1:hasMember>
    <aixm:RunwayDirectionLightSystem gml:id="F749">
      <gml:identifier codeSpace="urn:uuid">RDLS4</gml:identifier>
      <aixm:timeSlice>
        <aixm:RunwayDirectionLightSystemTimeSlice gml:id="TS749">
          <gml:validTime>
            <gml:TimePeriod gml:id="VT750">
              <gml:beginPosition>2010-10-
01T08:59:00.000Z</gml:beginPosition>
              <gml:endPosition
indeterminatePosition="unknown"></gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:timeSliceMetadata/>
          <aixm:element>
            <aixm:LightElement gml:id="id751">
              <aixm:colour>RED</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P752">
                  <gml:pos>-87.93345173677841 41.99031071893752</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
                <aixm:LightElementStatus>
                  <aixm:status>NORMAL</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
          <aixm:element>
            <aixm:LightElement gml:id="id751">
```

```
              <aixm:colour>GREEN</aixm:colour>
              <aixm:location>
                <aixm:ElevatedPoint srsName="urn:ogc:def:crs:OGC:1.3:CRS84"
gml:id="P752">
                  <gml:pos>-87.93343717040139 41.990296152560504</gml:pos>
                </aixm:ElevatedPoint>
              </aixm:location>
              <aixm:availability>
                <aixm:LightElementStatus>
                  <aixm:status>NORMAL</aixm:status>
                </aixm:LightElementStatus>
              </aixm:availability>
            </aixm:LightElement>
          </aixm:element>
        </aixm:RunwayDirectionLightSystemTimeSlice>
      </aixm:timeSlice>
    </aixm:RunwayDirectionLightSystem>
  </ns1:hasMember>
</ns1:AIXMBasicMessage>
```

Annex C - Change Request - SE Symbolizer and Rule for styling of nested child objects

1 Source

OWS-8-Aviation

2 Category

Addition of feature

3 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.

4 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>
  <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 CompositeRule? we could in the same way define a rule that will only use the color from the child if it is not null. A composite rule extends a regular rule, so it can also have a filter to determine whether any of its contained rules have to be checked. It also has a flag to indicate whether the contained block of rules should be treated as an 'else if' block. An extensive example of composite rules can be found here: http://wrs1.galdosinc.com/ows8/query?request=GetRepositoryItem&id=urn:luciad:com:sld:runwaydirectionlightsystem

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:CompositeCh
ildSymbolizerType"
    substitutionGroup="se:Symbolizer">
  <xsd:annotation>
    <xsd:documentation>A "CompositeChildSymbolizer" comprises a g
roup 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="CompositeRule" type="ows:CompositeRuleType" su
bstitutionGroup="se:Rule"/>

<xsd:complexType name="CompositeRuleType" >
  <xsd:annotation>
```

```
    <xsd:documentation>
      Composite rules group a set of other rules that can be appl
ied to an object or a list of sub-objects.
      A composite rule can also have a filter to determine whethe
r its containing rules should be evaluated.
      The composite rule does not change the semantics of the pro
perties inherited from rule.
      If the 'ChildProperty' is set, all rules and symbolizers ar
e applied to the object(s) referenced by this
      property.
      The 'elseIfRule' attribute determines the semantics of how
the set of rules should be applied to the object(s).
      When set to 'true', only the first matching rule is applied
. When set to 'false', the semantics are the same as
      those used for FeatureTypeStyles.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="se:RuleType">
      <xsd:sequence>
        <xsd:element ref="se:Rule" maxOccurs="unbounded"/>
        <xsd:element name="ChildProperty" type="ogc:PropertyNameT
ype"/>
      </xsd:sequence>
      <xsd:attribute name="elseIfRule" type="xsd:boolean" default
="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

5 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 cases.