

Paper for Consideration by TSMAD/DIPWG

Potential Adjustments to S-100 Part 9 Portrayal - Text Styles.

Submitted by:	CARIS
Executive Summary:	This paper discusses the introduction of named Text Styles in S-100 Part 9 Portrayal
Related Documents:	S-100 Part 9 Portrayal
Related Projects:	S-101, S-100 Based Product Specifications

Introduction / Background

CARIS has been investigating the migration of S-52 presentation lookup table rules into S-100 XSL templates. With something like 284 text instructions using 46 distinct patterns there is only one text style being used. CARIS is proposing that the text styles be defined as registered catalogue items that would be stored in the Portrayal Register. Before going ahead with making redline changes and S-100 Part 12 proposals CARIS would like some consensus from the group or alternative approaches regarding this topic.

Analysis/Discussion

In S-100 Part 9 the Text package defines the types used for the depiction of text. In the current spec each text element carries with it a set of parameters used to define the style of the text. Namely the Font information (weight, slant and either a Font name or serifs and proportion), bodysize, text Flags, vertical Offset, foreground and optional background colour.

Sample Text Point Instruction

```
<textPoint>
  <element>
    <text>by Eaton</text>
    <bodySize>10</bodySize>
    <foreground>CHBLK</foreground>
    <fontCharacteristics>
      <weight>Medium</weight>
```

```

        <slant>Upright</slant>
        <serifs>0</serifs>
        <proportion>MonoSpaced</proportion>
    </fontCharacteristics>
</element>
<offset>
    <x>-3.51</x>
    <y>3.51</y>

</offset>
</textPoint>

```

The proposal is to separate the element style information into a named element that can be referenced by the Text instruction. To reproduce S-52 only one text style is needed since all the text is size 10 CHBLK.

Proposed Text Style Definition

```

<textStyle id="default">
    <bodySize>10</bodySize>
    <foreground>CHBLK</foreground>
    <fontCharacteristics>
        <weight>Medium</weight>
        <slant>Upright</slant>
        <serifs>0</serifs>
        <proportion>MonoSpaced</proportion>
    </fontCharacteristics>
</textStyle>

```

Proposed Text Point Instruction using Text Style reference

```

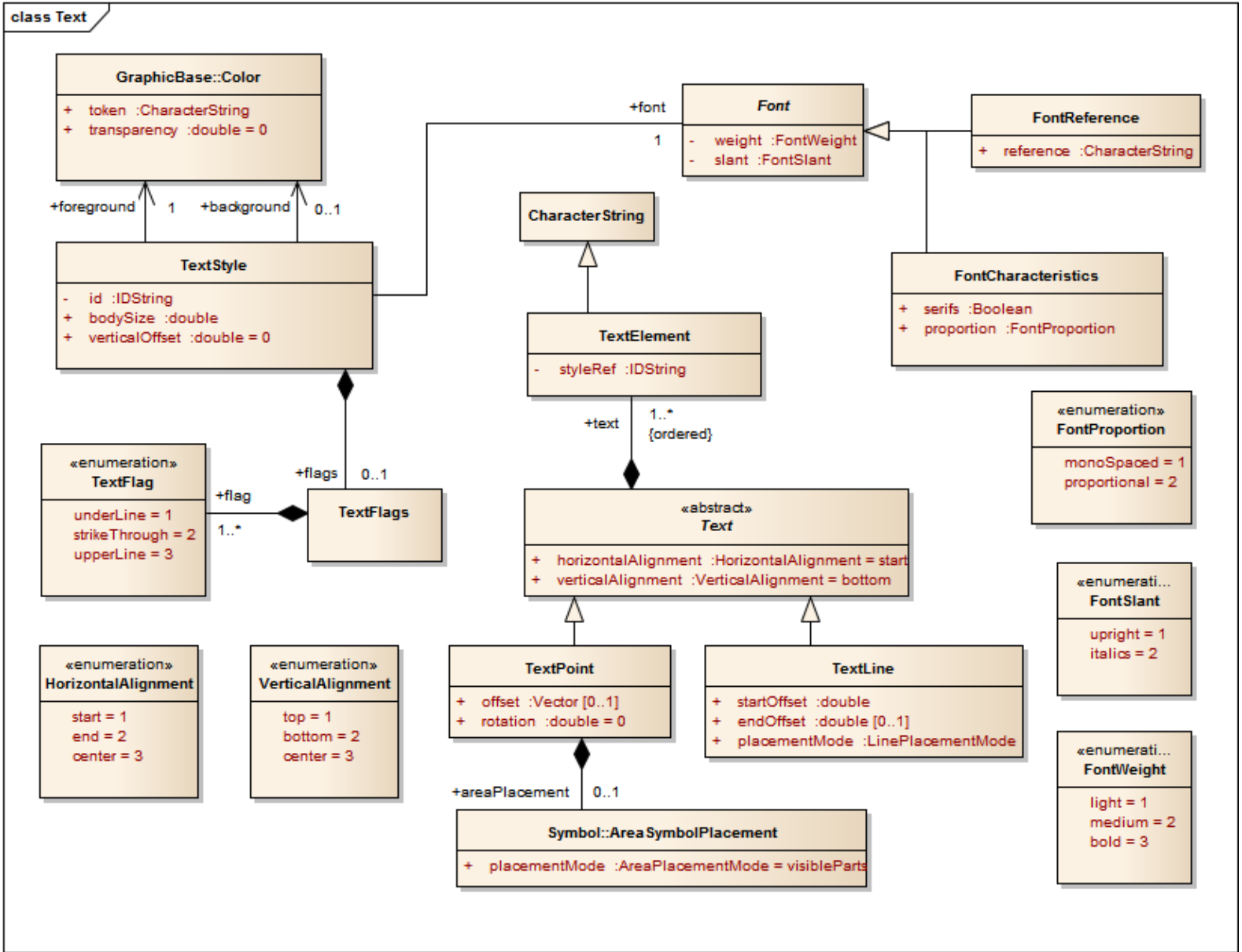
<textPoint>
    <text styleRef="default">by Eaton</text>

```

```
<offset>  
  <x>-3.51</x>  
  <y>3.51</y>  
</offset>  
</textPoint>
```

The proposed solution will reduce the number of elements that need to be populated when defining a text instruction and will reduce the size and complexity of the text instruction in the XML form. It should also facilitate maintenance when it comes to changing a text style. If the font or another of the style parameters needs to be changed it could be changed in the style definition instead of having to find and change every applicable text instruction in the portrayal catalogue.

The updated UML for the Text package could look like this:



Alternative, work around for having no style definition

Without having a text style definition the XSLT capabilities can be leveraged by using an XSL named template for the population of the text style.

```
<textPoint>
  <element>
    <xsl:apply-templates select="featureName/name" mode="text">
      <xsl:with-param name="prefix">by </xsl:with-param>
    </xsl:apply-templates>
    <xsl:call-template name="textStyle">
      <xsl:with-param name="style">default</xsl:with-param>
    </xsl:call-template>
  </element>
  <offset>
    <x>-3.51</x><y>3.51</y>
  </offset>
</textPoint>
```

The textStyle template would look something like this:

```
<!-- Template for text style elements of a text instruction -->
<xsl:template name="textStyle">
  <xsl:param name="style" select="default"/> <!-- set to 'default' if param not given -->

  <xsl:choose>
    <xsl:when test="$style = 'default'">
      <xsl:element name="bodySize">10</xsl:element>
      <xsl:element name="foreground">CHBLK</xsl:element>
      <xsl:element name="fontCharacteristics"> <!-- The font -->
        <xsl:element name="weight">Medium</xsl:element>
        <xsl:element name="slant">Upright</xsl:element>
        <xsl:element name="serifs">0</xsl:element> <!-- boolean no serifs -->
        <xsl:element name="proportion">MonoSpaced</xsl:element>
      </xsl:element> <!-- The font -->
    </xsl:when>
    <xsl:otherwise>
```

```
<xsl:text>error:unrecognized text style parameter: <xsl:value-of select="$style"/></xsl:text>  
</xsl:otherwise>  
</xsl:choose>
```

```
</xsl:template>
```

This template solution may work but it is not the preferable option. Using this template solution would not give a set of registered text styles and would not be consistent with how linestyles etc have been modelled.

Conclusions

As noted above, the introduction of a text style element should be considered as an improvement to the S-100 Part 9 Portrayal.

Action Required of TSMAD/DIPWG

The TSMAD/DIPWG is invited to:

- a. discuss the proposed text style as described in this paper
- b. record any follow on actions for the S-100 working group.