Features Needed in SVG2 to Support Technical Diagrams

Thomas Smailus
February 2013

Capturing the Boeing Use Cases

In defining what features SVG2 should have, given that many of Boeing's technical diagrams are published as CGM and other formats, which support some special features, we are attempting to understand the use of, in current diagrams, of certain features. Under consideration for feature addition are the following items:

1. textHeight (analogous to textLength)
2. Non-scaling line width
3. Non-scaling dash patterns
4. Non-scaling fill/hatch patterns
5. Engineering line types
6. Line style definition

The following sections describe the issues relative to each of these items in greater detail. If the reader has specific example diagrams that make use of the features in question, please contact the author.

textHeight

The existing implementation in SVG 1.1 of the use of *font-size* is sufficient to satisfy the CGM 1 and CGM 3 'basic' method of restricted text which only require the resulting text fill but not exceed the bounding box.

If controlling restricted text precisely is critical to the end user, i.e. the CGM 1 definition of restricted text is not sufficient, but more precise control of restricted text rendering as the CGM 3 standard allows (via Restricted Text Type), then SVG will need a more precise way of constraining the height of text than *font-size*.

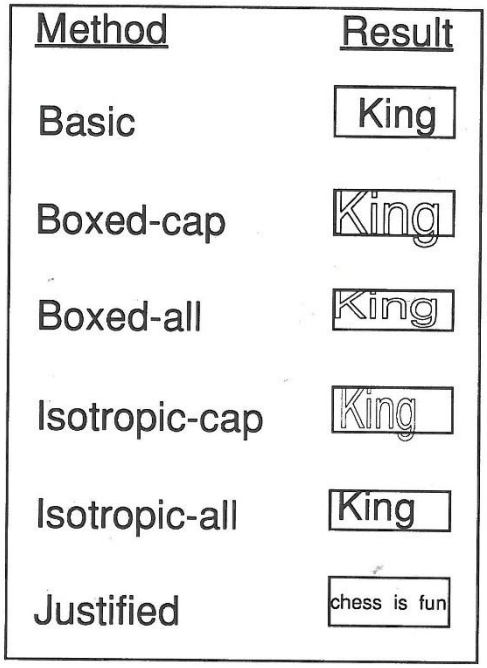

Figure 2. CGM Restricted Text Type definitions and their effects[1]

To exactly specify the stroked height of the glyphs from base to cap, specifying a font-size is an approximation because font-size includes cap to top and base to bottom.

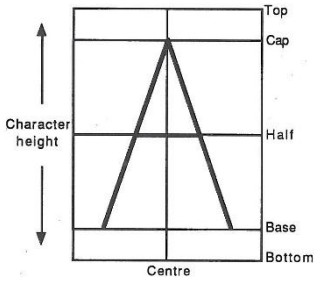

Figure 1. CGM terminology with respect to text glyphs[1]

textHeight with textLength in conjunction with length-Adjust and height-Adjust would provide a mechanism for implementing CGM Restricted Text (ISO/IEC 8632-

1:1999(E) 7.6.5) in CGM 3 files with restriction methods (Restricted Text Type). The types CGM 3 supports are shown in the accompanying figure[1].

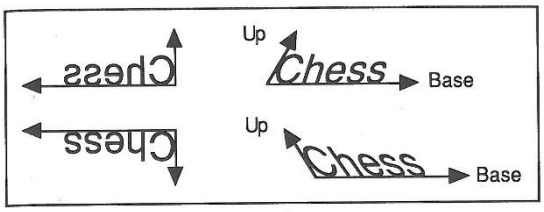

Figure 3. CGM text vectors and their effect on text[1]

Restricted Text also allows for the definition of a parallelogram to restrict the text, due to the ability of CGM text glyph transformations to be specified in greater detail by the use of UP and BASE vectors. The use of non-rectangular restriction boxes (the right side cases in the accompanying figure[1]) is suspected to be highly limited in the diagrams we have worked with, and a use case does not yet exist for adding support for these. Allowing for only rectangular bounding boxes (the 2 left side cases in the accompanying figure[1]), implemented with textHeight and textWidth should be sufficient.

Non-scaling line width

Users of large diagrams with many details will need to zoom in on details. This is done by adjusting the scaling of the diagram, either in the SVG document, or by the rendering engine of the viewer. There is an engineering drawing need for a line style that keeps the same line width as rendered, regardless of zoom factor. In the current case, if one zooms in by a factor of 2x then the line thicknesses render 2x width on the display. Non-scaling line width would render the line with the same line width as in the 1x zoom case. This is analogous to the CGM *Line Width Specification Mode* values of scaled (default & how SVG 1.2 works) and absolute (the mode needed in SVG 2).

This figure illustrates the effect. Two horizontal lines drawn: the top on in *scaled* mode and the bottom one in *absolute* mode. If one zooms in by a factor of 2x, the scaled line becomes thicker, if one zooms out by a factor of 2x the scaled line becomes thinner. The absolute line maintains its thickness.

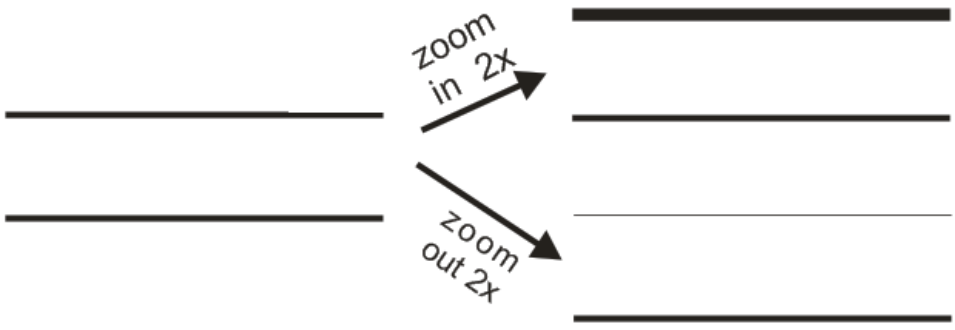

Figure 4. Effects of zooming on line width

CGM (ISO/IEC 8632-1:1999(E)) defines the behavior of *line width* to be

> if LINE WIDTH SPECIFICATION MODE is ‘absolute', 1/1000 of the longest side of the rectangle defined by default VDC EXTENT;
> if LINE WIDTH SPECIFICATION MODE is ‘scaled', 1.0;
> if LINE WIDTH SPECIFICATION MODE is ‘fractional', 0.001;
> if LINE WIDTH SPECIFICATION MODE is ‘mm', 0.35

[1] Figures from Henderson, Mumford, “The CGM Handbook,” Academic Press, 1993

What line width should be in a non-scaling *absolute* mode in SVG would need to be determined. One option is to render according to how the SVG is displayed by *viewBox* attribute of the *SVG* element, and keep that size through all scale setting.

Non-scaling dash patterns

Just as with line width, there needs to be an option for not scaling the dashed patterns, referred to here as an *absolute* mode.. As with the line width example, the dashed example contains 2 lines, the top one in *scaled* mode, and the bottom one in *absolute* mode. When zooming in, the scaled line's pattern also scales up (with the line thickness). When zooming out, the scaled line's pattern scales down. The absolute line's patter does not change pattern, regardless of zoom level. Note that the line width is also zoom vs. absolute in this case.

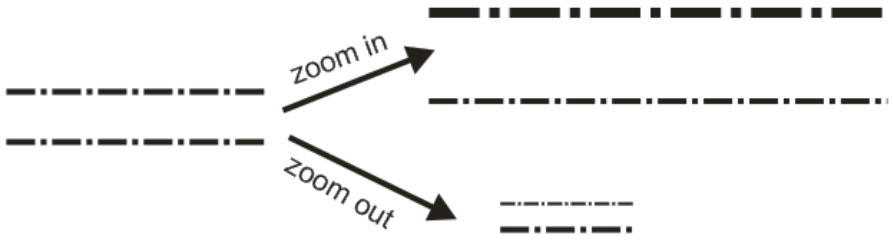

Figure 5. Effects of zooming on line width and dashed patterns

Non-scaling fill/hatch patterns

Just as with line width, there needs to be an option for not scaling the fill patterns, referred to here as an *absolute* mode. Two rectangles filled with a hatch pattern on the left of the figure. The upper one is in *scale* mode, the lower one in *absolute* mode. As one zooms in, the upper pattern changes with the zoom level, while the *absolute* mode pattern stays the same. As one zooms out, the same holds true, the upper *scaled* mode pattern gets smaller, while the *absolute* mode pattern stays the same.

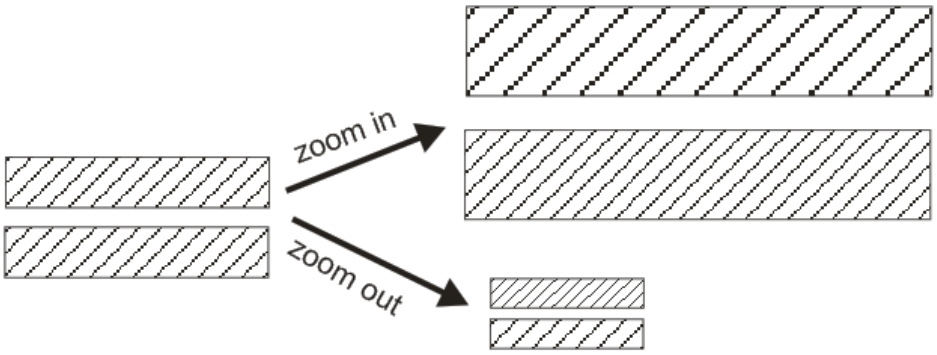

Figure 6. Effect of zooming on hatch fill patterns

Engineering Line Types

ISO 128 defines principles of technical drawings and how things are graphically represented. We need to understand the value and usage of the various engineering line types and decide if there needs to be

native support for the line types in the graphics standard, or at least by some extension. The following image[2] of the ISO 128 Type A through Type K describe the appearances. Many of the types look like they can be rendered using the dash pattern and line width capabilities of SVG 1.1. Types C, D, and H are more unique. They might be rendered via a rendering algorithm in ECMAscript embedded in the diagrams as a function call. Such programmatically drawn lines – would they be properly affected by attribute inheritance or transformations applied to the ancestors, such as diagram panning and zooming?

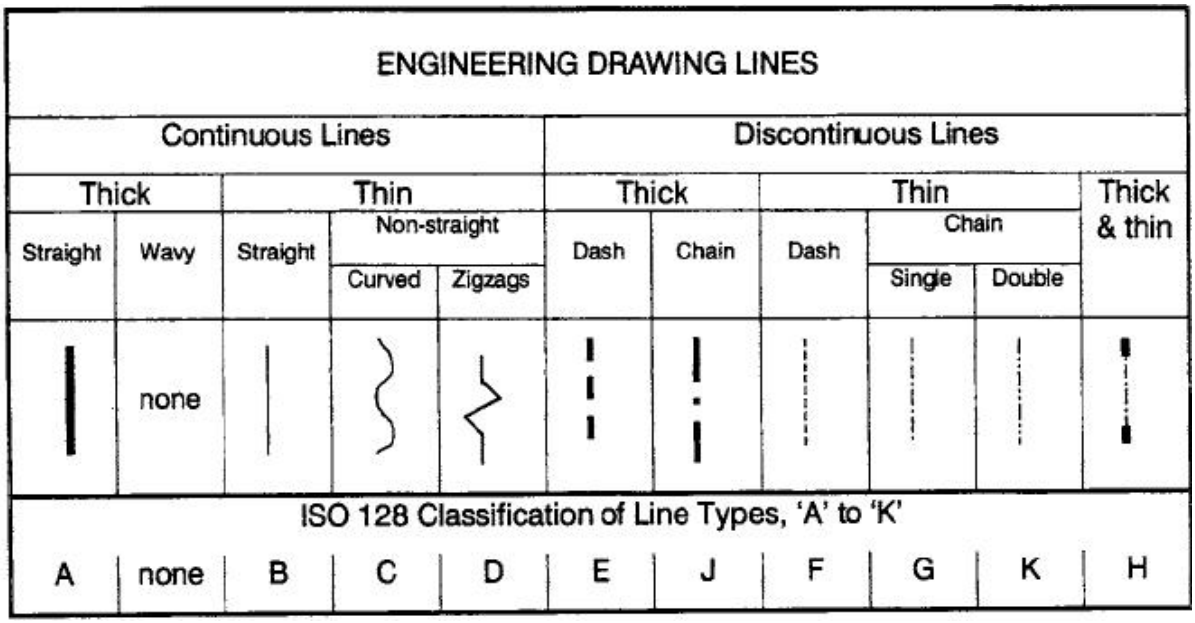

Figure 7. ISO 128 Classification of Line Types[2]

AutoCad, a very popular computer drafting application, provides libraries of line types:

- 24 standard line types
  - 8 Basic Patterns
    - Border, center, dash-dot, dashed, divide, dot, hidden, phantom
  - Each of these in 3 spacing variants: ½ , normal, 2x
- 14 ISO 128 line types conforming to the ISO/DIS 12011 specification.
  - These line types may be rarely used. Need to investigate
- 7 complex line types
  - Include symbols or text, dash-dot line patterns, fence line, railroad tracks, batt insulation, etc.

[2] Image from *http://imeulia.blogspot.com/2012_11_01_archive.html* on February 7, 2014

The following figure[3] depicts representations and uses of various engineering line types in an engineering drawing.

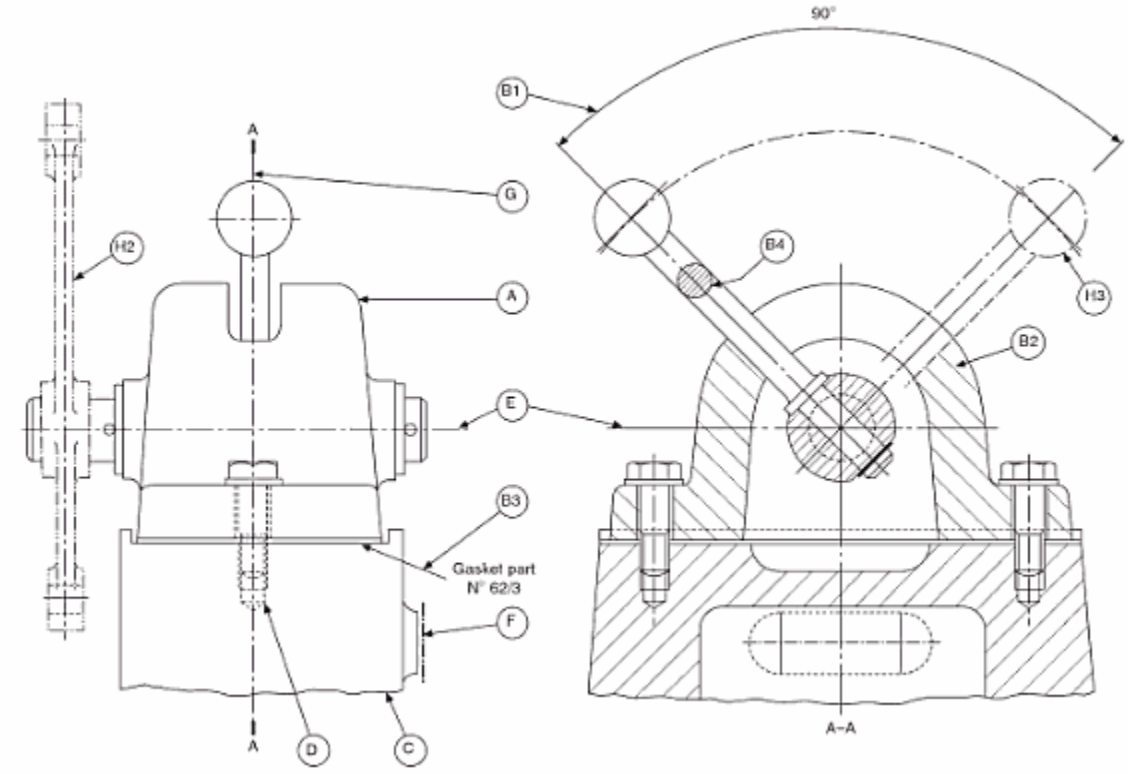

Figure 8. Sample use of Engineering Line Types[1]

Line Style Definition

CGM defines several methods of handling dashed line patterns at internal vertices and corners (ISO/IEC 8632-1:1999(E) 6.7.1.2 c) via LINE TYPE CONTINUATION:

Unspecified	no specific treatment is required;
Continue	the style is continued without interruption across vertices;
Restart	the style is restarted at each vertex;
Adaptive continue	the style is continued, but each vertex must be "inked" including vertices at the ends of the line primitive which might otherwise not be drawn because of a non-solid line type.

Is there a need for these 4 behaviors in engineering drawings? If so, then a *line type continuation* equivalent would be needed for SVG.

[3] Simmons, Phelps, Maguire, "Manual of Engineering Drawing, Fourth Edition: Technical Product Specification and Documentation to British and International Standards," Butterworth-Heinemann, 4th Edition, 2008, p. 55