Stage 3 proposal: Feature 13119 SVG domain

Contents

Stage 3 proposal: Feature 13119 SVG domain

Official domain for SVG

Champion

Eliot Kimber

Tracking information

Event	Date	Links
Stage 1 proposal accepted	25 Sept 2012	*https://www.oasis-open.org/committees/document.php?document_id=47001&wg_abbrev=dita*
Stage 2 proposal submitted	1 Oct 2012	*https://www.oasis-open.org/apps/org/workgroup/dita/download.php/47062/proposal-13119-dita-source.zip*, *https://www.oasis-open.org/apps/org/workgroup/dita/download.php/47063/proposal-13119-html.zip*
Stage 2 proposal discussed	16 Oct 2012	*https://www.oasis-open.org/apps/org/workgroup/dita/download.php/47205/minutes20121016.txt*
Stage 2 proposal approved	30 October 2012	
Stage 3 proposal submitted to reviewers	2 July 2013	Debra Bissantz
Stage 3 proposal (this document) submitted		

Approved technical requirements

Define a new vocabulary module, svgDomain, that defines the following element type:

- <svg_container>

  Specializes topic/foreign. Allows as content the <svg:svg> element from the SVG 1.1 vocabulary, <data>, or <data-about>, as a repeating OR group.

  Because the SVG vocabulary includes some elements that have the same local name as DITA elements, SVG included in DTD-based or RelaxNG-based documents must use prefixed tagnames. Documents that use XSD schemas may use unprefixed tagnames.
- Include the SVG 1.1 DTD declarations. See *http://www.w3.org/TR/SVG11/svgdtd.html*.

  There do no appear to be normative XSD or RelaxNG versions of the SVG vocabulary.

Dependencies or interrelated proposals

No related proposals.

Modified DTDs

svgDomain.ent

Create new declaration set `svgDomain.ent` in the base declaration set package with the following content:

```
<?xml version="1.0"
 encoding="utf-8"?>
<!--
 ============================================
     DITA SVG Domain

     Defines a specialization of
 <foreign> that contains
     SVG markup.

     DITA 1.3

     Copyright (c) 2012, 2013 OASIS
 Open


 ============================================
 -->


<!--
 ============================================
 -->
<!--                    SVG DOMAIN
 ENTITIES                    -->
<!--
 ============================================
 -->

<!-- SVG elements must be prefixed
 when validated by DTD, otherwise
 they
     conflict with existing
 DITA elements (e.g., <desc> and
 <title>).
  -->
<!ENTITY % NS.prefixed "INCLUDE" >
<!ENTITY % SVG.prefix "svg" >

<!ENTITY % svg-d-foreign
   "svg_container
   "
>

<!ENTITY   svg-d-att
  "(topic svg-d)"
>

<!-- ================== End DITA SVG
 Domain Entities ============ -->
```

svgDomain.mod

Create new declaration set file `svgDomain.mod` with the following content:

```
<?xml version="1.0"
 encoding="utf-8"?>
```

```
<!--
 ============================================
     DITA SVG Domain

     Defines a specialization of
 <foreign> that contains
     SVG markup.

     DITA 1.3

     Copyright (c) 2012, 2013 OASIS
 Open


 ============================================
 -->

 <!ENTITY % svg_container
 "svg_container" >
 <!ENTITY % svgref
 "svgref" >

<!ENTITY % svg11.dtd
  SYSTEM "svg11/svg11.dtd"
>%svg11.dtd;

<!--
 ============================================
 -->
<!--                    ELEMENT NAME
 ENTITIES                          -->
<!--
 ============================================
 -->


<!--
 ============================================
 -->
<!--                     ELEMENT
 DECLARATIONS
 -->
<!--
 ============================================
 -->

<!ENTITY % svg_container.content
"
  (%data; |
   %data-about; |
   %SVG.pfx;svg)*
"
>
<!ENTITY % svg_container.attributes
 "
  %id-atts;
  %localization-atts;
  base
    CDATA

    #IMPLIED
  %base-attribute-extensions;
  outputclass
```

```
    CDATA

    #IMPLIED

 "
>
<!ELEMENT svg_container
 %svg_container.content; >
<!ATTLIST svg_container
 %svg_container.attributes; >

<!-- LONG NAME: SVG Reference -->
<!ENTITY % svgref.content
"EMPTY"
>
<!ENTITY % svgref.attributes
            "href
                        CDATA

 #IMPLIED
             keyref
                        CDATA

 #IMPLIED
             type
                        CDATA

 #IMPLIED
             format
                        CDATA

 "svg"
             scope
                        (external |
                         local |
                         peer |
                         -dita-use-
conref-target)

 #IMPLIED
             %univ-atts;
             outputclass
                        CDATA

 #IMPLIED"
>
<!ELEMENT svgref
 %svgref.content;>
<!ATTLIST svgref
 %svgref.attributes;>

<!--
 ============================================
 -->
<!--
 SPECIALIZATION ATTRIBUTE
 DECLARATIONS      -->
<!--
 ============================================
 -->

<!ATTLIST svg_container
   %global-atts;  class CDATA "+
```

```
 topic/foreign svg-d/svg_container
 ">
<!ATTLIST svgref
   %global-atts;  class CDATA "+
 topic/xref    svg-d/svgref ">

<!-- ================== End SVG
 Domain ==================== -->
```

Modified specification documentation

Table 1: New and Modified Topics

Topic File	DITA 1.2 Text	DITA 1.3 Text	
common/svg-d.dita		New topic, under Domain elements	
langref/svg_container.dita		New topic	
langref/svgref.dita		New topic	
archSpec/dita-linking-link-element-summary.dita		Add an entry for <svgref>	
		svgref (SVG domain)	Uses an SVG <svg> element stored in a non-DITA document by reference.

SVG elements

The SVG domain elements enable direct use of SVG markup within DITA documents, as well as use-by-reference of SVG markup held in separate non-DITA documents. SVG is a W3C standard, http://www.w3.org/TR/SVG/. Note that for SVG markup stored directly in DTD-validated DITA documents, the SVG elements must use a namespace prefix to avoid conflict with DITA-defined elements of the same name. Documents validated using XSD or RelaxNG may default the SVG namespace on the SVG <svg> element. SVG elements used by reference using the <svgref> element do not need to have a namespace prefix because they will be parsed separately from the DITA documents that refer to them. The domain is configured by default to use the namespace prefix "svg" for the SVG elements.

svg_container

Contains SVG elements, references to SVG elements held in separate, non-DITA documents, <data>, or <data-about>. The SVG markup must have a root element of "svg" within the SVG namespace "http://www.w3.org/2000/svg".

Inheritance

+ topic/foreign svg-d/svg_container

Example

SVG containers with inline SVG markup:

```
<topic id="svg-test-topic-01">
  <title>SVG Domain Test: Namespace Prefixed SVG Elements</title>
  <body>
    <p>SVG Inline: <svg_container>
      <svg:svg
        width="100"
        height="100">
```

```
        <svg:defs>
          <svg:filter
            id="f1"
            x="0"
            y="0">
            <svg:feGaussianBlur
              in="SourceGraphic"
              stdDeviation="15"/>
          </svg:filter>
        </svg:defs>
        <svg:rect
          width="90"
          height="90"
          stroke="green"
          stroke-width="3"
          fill="yellow"
          filter="url(#f1)"/>
      </svg:svg>
      </svg_container></p>
    <p>SVG Directly in body:</p>
    <svg_container>
      <svg:svg width="200" height="200">
        <svg:ellipse cx="100" cy="100" rx="80" ry="80" style="fill:blue;
          stroke:rgb(0,0,100);stroke-width:2"/>
      </svg:svg>
    </svg_container>
    <fig>
      <title>Figure With SVG Container</title>
      <svg_container>
        <svg:svg width="4in" height="6in" version="1.1"
          >
          <svg:circle cx="150" cy="200" r="100" fill="url(#grad_blue)" >
            <svg:animate attributeName="r" begin="Go.click" end="Stop.click"
 dur="4s"
              values="100; 0; 100" repeatCount="indefinite"/>
          </svg:circle>
          <svg:radialGradient id="grad_blue" cx="20%" cy="20%" r="100%"
 fx="30%" fy="30%">
            <svg:stop stop-color="white" offset="0"/>
            <svg:stop stop-color="blue" offset="25%"/>
            <svg:stop stop-color="rgb(0,0,192)" offset="50%"/>
            <svg:stop stop-color="rgb(0,0,127)" offset="70%"/>
            <svg:stop stop-color="rgb(0,0,64)" offset="85%"/>
            <svg:stop stop-color="rgb(0,0,0)" offset="100%"/>
          </svg:radialGradient>
          <svg:g id="Go">
            <svg:rect x="70" y="320" height="40" width="80" fill="aqua"/>
            <svg:text x="90" y="350" font-size="30" fill="green">Go </
svg:text>
          </svg:g>
          <svg:g id="Stop">
            <svg:rect x="160" y="320" height="40" width="80" fill="aqua"/>
            <svg:text x="170" y="350" font-size="30" fill="red">Stop</
svg:text>
          </svg:g>
        </svg:svg>
      </svg_container>
    </fig>
  </body>
</topic>
```

An SVG container with a reference to SVG markup held in a separate non-DITA document:

```
<fig>
  <title>Figure With SVG Container</title>
  <svg_container>
    <svgref keyref="svg-fragment-01"/>
  </svg>
</fig>
```

Where the key "svg-fragment-01" is declared like so:

```
<map>
 ...
 <keydef keys="svg-fragment-01"
   href="media/svg/svg-library.xml#frag-0001"
   format="svg"
 />
 ...
</map>
```

Attributes

Generated attribute table goes here.

svgref

Refers to a non-DITA XML document containing SVG markup in order to use the markup by reference.

The reference must be to an SVG <svg> element. The reference may be a URI that addresses an XML document whose root element is an SVG <svg> element or a URI that addresses an XML document and a fragment identifier that is the XML ID of an <svg> element within the document.

Note: To reuse SVG markup stored within a DITA topic, use a normal content reference from the <svg_container> element.

The SVG should be processed and rendered as though the <svg> element had occurred directly in the content of the containing <svg_container> element.

The reference may be direct, via the @href attribute, or indirect, via the @keyref attribute.

For key references, only the key name should be specified. Any fragment identifier for specifying the ID of the <svg> element to use must be specified as part of the key definition's @href value.

Example:

To refer to the <svg> element with the @id value "svg-fragment-02" within a larger document using a key reference, you would define the key like so:

```
<keydef
  keys="svg-fragment-0002"
  href="svg/svg-library.xml#svg-fragment-02"
  format="svg"
/>
```

You would refer to this key using just the key name:

```
<svg_container>
  <svgref keyref="svg-fragment-0002"/>
</svg_container>
```

Inheritance

+ topic/xref svg-d/svgref

Example

A reference to an <svg> element that is the root element of its containing document:

```
<fig>
  <title>Figure With SVG Container</title>
  <svg_container>
    <svgref
       href="media/svg/svg-graphic-01.xml"
       format="svg"
    />
  </svg>
</fig>
```

`svg-graphic-01.xml` (note that the <svg> element sets the SVG namespace as the default namespace, so there are no namespace prefixes on the SVG markup):

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg"
  width="100"
  height="100">
  <defs>
    <filter
      id="f1"
      x="0"
      y="0">
      <feGaussianBlur
        in="SourceGraphic"
        stdDeviation="15"/>
    </filter>
  </defs>
  <rect
    width="90"
    height="90"
    stroke="green"
    stroke-width="3"
    fill="yellow"
    filter="url(#f1)"/>
</svg>
```

A reference to an <svg> element by ID:

```
<fig>
  <title>Figure With SVG Container</title>
  <svg_container>
    <svgref
      href="media/svg/svg-library.xml#frag-0001"
    />
  </svg>
</fig>
```

`svg-library.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <part>
   <svg id="frag-0001"
     xmlns="http://www.w3.org/2000/svg"
```

```
     width="100"
     height="100">
     <defs>
        <filter
          id="f1"
          x="0"
          y="0">
          <feGaussianBlur
            in="SourceGraphic"
            stdDeviation="15"/>
        </filter>
      </defs>
      <rect
        width="90"
        height="90"
        stroke="green"
        stroke-width="3"
        fill="yellow"
        filter="url(#f1)"/>
    </svg>
    <svg id="frag-0002"
       width="4in" height="6in" version="1.1"
          >
          <circle cx="150" cy="200" r="100" fill="url(#grad_blue)" >
            <animate attributeName="r" begin="Go.click" end="Stop.click"
 dur="4s"
              values="100; 0; 100" repeatCount="indefinite"/>
          </circle>
          <radialGradient id="grad_blue" cx="20%" cy="20%" r="100%" fx="30%"
 fy="30%">
            <stop stop-color="white" offset="0"/>
            <stop stop-color="blue" offset="25%"/>
            <stop stop-color="rgb(0,0,192)" offset="50%"/>
            <stop stop-color="rgb(0,0,127)" offset="70%"/>
            <stop stop-color="rgb(0,0,64)" offset="85%"/>
            <stop stop-color="rgb(0,0,0)" offset="100%"/>
          </radialGradient>
          <g id="Go">
            <rect x="70" y="320" height="40" width="80" fill="aqua"/>
            <text x="90" y="350" font-size="30" fill="green">Go </text>
          </g>
          <g id="Stop">
            <rect x="160" y="320" height="40" width="80" fill="aqua"/>
            <text x="170" y="350" font-size="30" fill="red">Stop</text>
          </g>
        </svg>
  </part>
</root>
```

Attributes

Generated attribute table goes here.