

**Awards, awards and more awards!**

UK Technical Communication, Horace Hockley and Mike Austin



# Communicator

The Institute of Scientific and Technical Communicators  
Winter 2017

Learn about Information 4.0

Create user-orientated  
documentation



**Eliminate  
Confusion**

We identify users' information needs and follow industry best practices to address them in a clear and easy-to-consume way.

Permanent and contract  
communicators.

"Because Edissero specialise in the field of technical communications, they see things that are easily overlooked by other recruiters. As a candidate, my dealings have been a pleasure. They have a relationship of trust, know how to market your skills, and prepare job interviews with care and it's easy to see why so many people's daily Edissero are the choice."

Do we need better authors?

Explore tools:  
SCHEMA ST4, SVG, Word



# SVG for multilingual procedural videos

In the fourth article on SVG, [Dave Gardiner](#) outlines the potential for producing simple videos with translated content in web-based help.

Scalable vector graphics (SVG), a mark-up language similar to HTML5, is gaining traction for producing high-quality illustrations and infographics in web documents. As you can add hyperlinks to illustrations, the format offers a basis for creating interactive visualisations. SVG, though, is mostly still perceived as a format where you should 'look but don't touch' – the capabilities of SVG for interactive technical documents are still vastly untapped, with many potential applications for interactive documents largely confined to small-scale software development projects. However, an existing online graphic design tool has recently been developed as a platform to take advantage of SVG's interactivity to enable new forms of technical documentation.

This article outlines how videos produced for technical documentation can be enhanced for global markets by adding SVG functionality to this graphics tool, so encouraging users to interact with documentation and offering another option for producing documentation.

## SVG integrates with HTML5

Gardiner (2014) outlined how SVG is integral to HTML5 and, when combined with cascading style sheets (CSS) and JavaScript, how it can enable the production of interactive visual web documentation. Further articles on SVG have focused on conceptualising visual-first interfaces for web documents (Gardiner 2015a), and testing the web accessibility of graphic elements with screen reader software and keyboard navigation (Gardiner 2015b). Applications for SVG are quite varied, and limited only by the multitude of customised JavaScript libraries and functions software that software developers integrate with SVG, to produce a range of dynamically generated, interactive and animated visuals in browsers.

Despite the diversity of innovative web development works by small teams of developers to make SVG graphics do more than just be another 'look-only' graphics format, current literature about SVG still typically focuses on integrating SVG, CSS and JavaScript to achieve visual design. Bellamy-Royds, Cagle and Storey (2016) explore applications of SVG with HTML5, CSS and JavaScript, but discussion centres on the styling of graphics and falls short of exploring one powerful element of SVG's markup, the <switch> element.

## Switching languages in hypergraphics

SVG mark-up comprises numerous elements and attributes controlling styling and behaviour of vector graphics in browsers, similarly to how HTML elements and attributes control the way textual content appears in browsers. In SVG, you can display parts of graphics on or off, as you can with HTML text. It is possible to draw illustrations in a graphics tool, export to SVG, then open the SVG in a text editor and add 'display: none' or 'visibility: hidden' attributes to any line, shape or text to prevent it being displayed in a browser. But this is an awkward way of treating SVG elements, such as text, because it means a user viewing a rendered graphic in a browser cannot interactively hide or display parts of it; text is either permanently displayed, or not, in this manner.

Taking the concept of displaying and hiding graphic elements with display or visibility attributes a step further for text content, SVG allows you to add translated text into a graphic, such as a technical illustration, and selectively display content for each language. The <switch> element allows control over displaying multilingual content that is in an SVG file. That is, you create a graphic and insert multiple translations of all related text within <switch>.

*SVG allows you to add translated text into a graphic, such as a technical illustration, and selectively display content for each language.*

```
<switch allowReorder="yes" transform="matrix(1 0 0 1 67.6011 237.4795)" fill="#ED1C24"
  class="overlay text">
  <text systemLanguage="en" font-size="24">Endnotes on separate page</text>
  <text systemLanguage="de" font-size="19">Setze Endnoten auf separate Seite</text>
  <text systemLanguage="it" font-size="15">Metti note di chiusura su una pagina
    separata</text>
  <text systemLanguage="af" font-size="20">Sit eindnote op 'n aparte bladsy</text>
  <text systemLanguage="fr" font-size="16">Mettez les notes sur une page
    séparée</text>
</switch>
```

Figure 1. SVG mark-up showing multilingual text contained in a <switch> element



Figure 1 shows the mark-up of part of an SVG file, where a <switch> element contains several <text> elements. Each <text> element includes the 'systemLanguage' attribute, which specifies the ISO 639-1 standard two-letter language code of the translated text. All major browsers support this language code standard. When selecting a preferred language in the browser settings, the text switches off the previously displayed language and switches on the selected language. Alternatively, you could show a list of languages in a graphic and a user taps one to display text content in their preferred language, thus circumventing the need to select a browser setting.

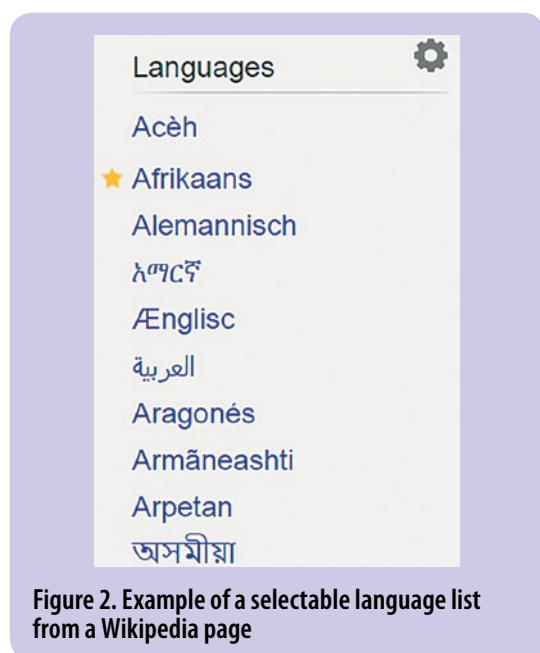


Figure 2. Example of a selectable language list from a Wikipedia page

The functionality operates in a similar way to multilingual websites, which show a list of languages in their native form. A user selects a preferred language and the web site switches the text content's language (Figure 2). And because SVG works with CSS, it is even possible to add CSS media queries so that a list of languages appears only when a browser is resized to the width of a mobile device.

### Technical videos and image-only instructions

Whilst videos are gaining acceptance as a form of technical documentation (Ellison 2017, Pierce 2014), they seem to be produced with commentary in specific languages. It is presumed that, for global markets, multiple copies of videos are produced with new voiceover or subtitles in the appropriate languages. This makes video production time-consuming because of the need to incorporate translated procedures as commentary for localisation.

Regarding visual documentation, techniques have been proposed to create visual-only instructions that do away with text completely –

such as the IKEA approach (Bayne 2007) – or at least minimise the amount of text with visuals (Thompson 2012).

### Putting videos and multilingual text together

I am proposing an approach that modifies the production of video documentation by combining the concepts of visual-only instructions with numbered steps in videos. The videos do away with any commentary and instead rely on numbered callouts that refer to translated content in web help.

It is a logical step to combine the multilingual capabilities of SVG with its ability to include video and audio. The SVG specification is designed to include HTML content, so SVG has had the ability to embed video within a graphic for many years. SVG mark-up has an element called <foreignObject>, which allows you to host videos inside a graphic, which in turn is contained within an HTML page. SVG is still somewhat a developer's environment because there are so many applications for the technology but so few tools with sophisticated WYSIWYG user interfaces (UI) that could automate this development coding work. It is not reasonable to expect a technical communicator to learn SVG coding to create media-based documentation.

### A production tool to achieve this

As I was scouring the web a few years ago looking for new SVG design tools, I came across Draw SVG. It looked promising because of the number of features available, but the online graphics tool was in poor shape. The user interface had an awkward design, with very small icons and overly technical jargon to describe interface features amongst its problems. I made some suggestions to the developer who, to my amazement, responded positively and was open to ideas for improvement.

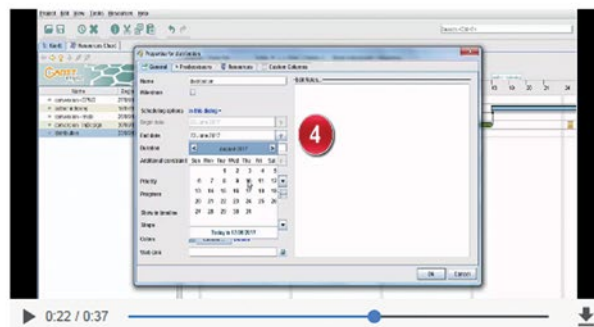
I have since worked closely with the developer to help improve the tool, and have taken it on as a 'pet project', contributing ideas for new features, and suggesting improvements to the UI and file bug reports. I realised this tool had to take

*It is a logical step to combine the multilingual capabilities of SVG with its ability to include video and audio.*

```
<html>
  <head>
    <title>Test linked SVG</title>
  </head>
  <body>
    <p lang="en">This SVG video is inside an HTML
      page:</p>
    <object data="video.svg" width="400" height="350"
      type="image/svg+xml">
      You cannot see SVG graphics.
    </object>
  </body>
</html>
```

Figure 3. HTML mark-up that refers to an SVG file using the <object> element

This translated SVG video is inside an HTML page:



1. Pour créer une tâche, cliquer avec le bouton droit de la souris puis sélectionner l'action "New task".
2. Nommer la tâche en éditant son champ "Name".
3. Double-cliquer sur l'icône verte de la tâche pour ouvrir son formulaire.
4. Définir sa date de fin à l'aide du calendrier.
5. Saisir sa durée.
6. Cliquer sur OK.
7. Faire glisser le calendrier pour atteindre le début de la tâche.
8. Pour définir son enchaînement, cliquer sur la fin de la tâche précédente puis faire glisser la flèche jusqu'au début de la tâche.

English Français 日本語

Figure 4. Multilingual video displayed in a web page

advantage of the powerful capabilities of SVG to differentiate it from other SVG graphic design tools, and struck upon the idea of using the `<switch>` element to create multilingual graphics.

With what was then rudimentary experience in working with agile software development teams, I wrote a basic 'functional specification' with mock-ups of the UI and dialog boxes, and the SVG coding necessary at each step when specifying languages and inserting translated text. The developer modified the design by adding essential JavaScript functions to the SVG coding to make the dialog boxes for user input, but the resulting new multilingual feature worked out well.

Adding translated text to a video embedded in an SVG drawing is achieved using this online tool without a user needing to see any coding. A video translation project in its simplest form need only involve copying translated procedures from a text file or Word document, pasting them into each language field in a dialog box, and styling the text for font style and aligning/indenting.

### Coding skills are required for styling content

If you are content to produce multilingual videos in this way, you need never see any SVG mark-up; you would only need to edit HTML mark-up to display an SVG document in a web page. Figure 3 shows HTML mark-up that refers to an external SVG file with the `<object>` element. But if you want to style illustrations or infographics that are included with videos, you need to dive into SVG mark-up.

Like most other SVG design tools, the style information for lines, shapes, fills and text – such as colours, line width and font size – is stored as attributes in the SVG coding. Styling of SVG follows a similar concept as HTML styling, where you can have styles specified as inline, internal or external cascading stylesheets (CSS). SVG styles generated from Draw SVG are

stored only as inline styles for each graphic element. To change styles means selecting each graphic element and picking new colours, etc. from dialog boxes.

If you want the flexibility of CSS to modify styles in an SVG file, you need to add each inline style to an external CSS and refer to it in the SVG (as you do with HTML) – a laborious process if the SVG contains a complex illustration or infographics with many elements. A recent improvement to Draw SVG has been to improve the process of creating an external CSS to replace inline styles.

### Compatibility with browsers and devices

While all major browsers allow you to display simple SVG illustrations, there are differences in how various desktop and mobile browsers allow interactive graphics to be displayed, for example, popup windows. Design of multilingual videos outlined in this article involves embedding a video at a fixed width and height within an SVG file that contains translated content and JavaScript to control language selection. Then you refer to the SVG from an HTML page so a user can access multilingual videos within web help formats.


With this file setup on Windows 7 desktop browsers (Chrome 60, Firefox 55) display the HTML page and allow users to switch languages both on-screen and through browser settings. Internet Explorer 11 does not display the video, yet does show text content, but you then cannot change languages on-screen. The expand icon is missing from the video controls bar, so you cannot expand the video to full-screen display. The video is displayed and functions similarly in Chrome 60 on Windows 10. Chrome 60 on Android 5 displays videos and allows a user to switch languages by tapping a language listed on-screen.

Viewing only the SVG file directly in a browser (that is, not linked from an HTML page) such as

Chrome and Firefox, allows you to expand the video to full screen, using the expand icon, that appears on the video controls bar.

Different setups produce differing results in different browsers. An example of this is embedding an SVG file within the HTML mark-up rather than linking from it. These varying results highlight an immediate need to develop complex interactive SVG documents (the HTML and SVG file setup) for specific browsers and operating systems on which users would access documentation. Enabling cross-browser support will be a further significant development issue for SVG.

### Acknowledgments

Thanks to Janet Taylor, Carl Sarelius and Corey Wallis of the Australian Society for Technical Communication for testing of multilingual videos across Windows versions and browsers. 

### References and further reading

- Bayne E (2007) 'Images only: where illustrations rule' *Communicator*, Summer 2007: 12-14
- Bellamy-Royds, A, Cagle, K and Storey, D (2016) *Using SVG with CSS3 and HTML5*, O'Reilly Media
- Ellison M (2017) 'Including videos in Flare's HTML5 output' *Communicator*, Summer 2017: 58-59
- Gardiner D (2014) 'Interactive hypergraphics design' *Communicator*, Winter 2014: 21-24
- Gardiner D (2015a) 'Visualising online help topics' *Communicator*, Spring 2015: 38-41
- Gardiner D (2015b) 'Accessibility of SVG hypergraphics' *Communicator*, Summer 2015: 38-41
- Pierce M (2014) 'Video for technical communicators' *Communicator*, Winter 2014: 30-31
- Thompson S (2012) 'Low-maintenance documentation techniques' *Communicator*, Summer 2012: 26-27

### Resources

Draw SVG by Joseph Liard, <http://drawsvg.org>

Examples of multilingual video are available at <http://svgdocs.net/drawsvg/gantt.svg>, and the same SVG file linked to an HTML page at <http://svgdocs.net/drawsvg/gantt-video.html> (accessed September 2017)



**Dave Gardiner MISTC** is an Australian technical communicator with expertise in XML, and a background in IT technical documentation and scientific illustration. He holds the tekomp

first-level certificate in technical communication.

E: [info@xmplar.biz](mailto:info@xmplar.biz)

W: [xmplar.biz](http://xmplar.biz)

T: @copeboox

## Be part of a network of exceptional communicators

Join the largest UK body representing information development professionals, and enjoy the resources and opportunities you need to be successful:

- Get the professional development and recognition you need
- Join communities that interest you
- Grow your professional network
- Share knowledge, join debates or start your own
- Collaborate to develop ideas and create content
- Learn and develop your skills
- Stay up to date with trends and technologies

[istc@istc.org.uk](mailto:istc@istc.org.uk)

[www.istc.org.uk](http://www.istc.org.uk)

