Translating SVG with XLIFF and Open Standards

Efficiently Solving a Classic Challenge

Bryan Schnabel

Information Architect
Tektronix, Inc.

Beaverton, Oregon, United States

<bschnabel@bschnabel.com>

Abstract

For years publishing groups have grappled with a daunting challenge. How do they ensure quality and consistency of translated text strings in images, and get translations for a reasonable price? Combine two critical developments and an answer emerges. Since **SVG** (Scalable Vector Graphics) files (`http://www.w3.org/Graphics/SVG/`) are XML, and since XML-based open standards are now in place to facilitate efficient translation, the challenge now has a solution.

In this presentation, an exact architecture and blue print will be provided that walks the attendees through real-world examples. Using the OASIS **XLIFF** (XML Localization Interchange File Format) 1.2 open standard (`http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff`), and SVG, a concrete solution will be demonstrated. The demonstrations show how the use of these two open standards increases quality and consistency of translation.

Using industry-supported open standards such as XLIFF allows localization service providers to focus on the translation of text, using localization tools (their core competency), rather than training translators to use complex graphic tools. This results in measurable savings in cost and time.

Attendees will see how traditional tools such as translation memory can be efficiently used when combining combing SVG with XLIFF. Additional standards for translation, localization, and internationalization will be explained as well, including:

- **LISA** (Localization Industry Standard Association, `http://www.lisa.org/Homepage.8.0.html`)
- W3C **ITS** (Internationalization Tag Set, http://www.w3.org/International/its/)

And attendees will be shown SVG graphics in technical documentation, in conjunction with powerful XML formats like:

- OASIS **DITA** (Darwin Information Typing Architecture, `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita`)
- W3C **XHTML** (Extensible HyperText Markup Language , `http://www.w3.org/TR/xhtml1/`)
- OASIS **Docbook** (`http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=docbook`)

The documentation, along with the graphics, will be translated in a standardized way. This concept will be shown using commercial and open source tools.

This is a dynamic, nut and bolts, fun presentation. Attendees experience a minimal amount of slides; learning is accelerated through the use of real-world examples and real-time demonstrations.

Table of Contents

The Classic Problem: Translating Text in Graphics

Technical publishers and LSPs (Localization Service Providers) have managed to translate text in graphics. But it has been problematic, expensive, and error prone.

Traditional Method 1: Manually Extract or Duplicate Strings - Translate, Copy, Paste

Translator or LSP Translates Text Strings - Publisher Pastes Strings Into Graphic File

Many publishing groups provide text strings to the translator in a word processor format, or spreadsheet file. And if necessary, or possible, they send along hard-copy pictures so the translators can see the context to aid in accurate translation.

Drawbacks of the Copy-and-Paste Method

This method often adds a lot of overhead for the project manager. Special care has to be taken to coordinate which strings go into which graphic files. It often requires the coordinated effort of three people, the translator (who understands the translated strings), the graphic artist (who can use the graphics software), and the technical writer (who oversees the operation to ensure proper page-layout).

This many participants in a process drives up cost and increases time. And to try to save money and time, sometimes groups attempt to use the copy-and-paste method with fewer participants. This often lead to errors and restarts.

Traditional Method 2: Requiring Translation of Proprietary Graphic Format Files

A common traditional way to translate text strings in graphics is to send the entire graphic file to the translator and pay to have the text translated.

Drawbacks of Having LSPs Translate Graphic Files

While many translators may be skilled at using proprietary graphic software, this means the customer is paying for more than just the translation. Also, care must be taken to ensure that the LSP and customer are each using the same version and OS graphics software.

Solving the Classic Problem with Open Standards

Several open standards can be used to solve this problem. The two primary standards are SVG and XLIFF.

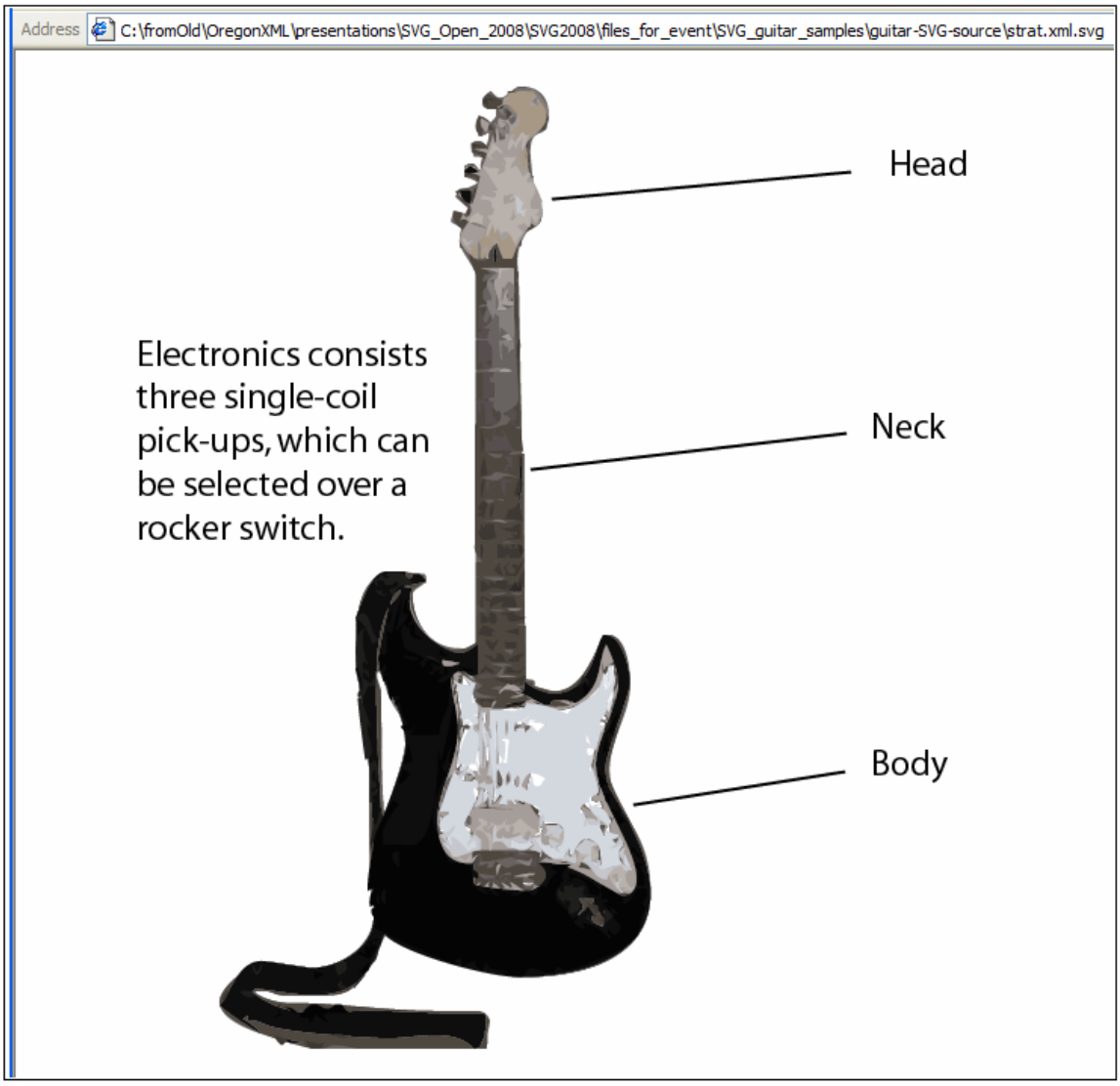
The Open Standards

SVG

SVG is an XML language for describing two-dimensional graphics and graphical applications. Its strength is that while SVG files can be easily rendered, created, and edited in several commercial and open source software packages, the fact that SVG files are XML means the text strings are easily identified, extracted, and transformed.

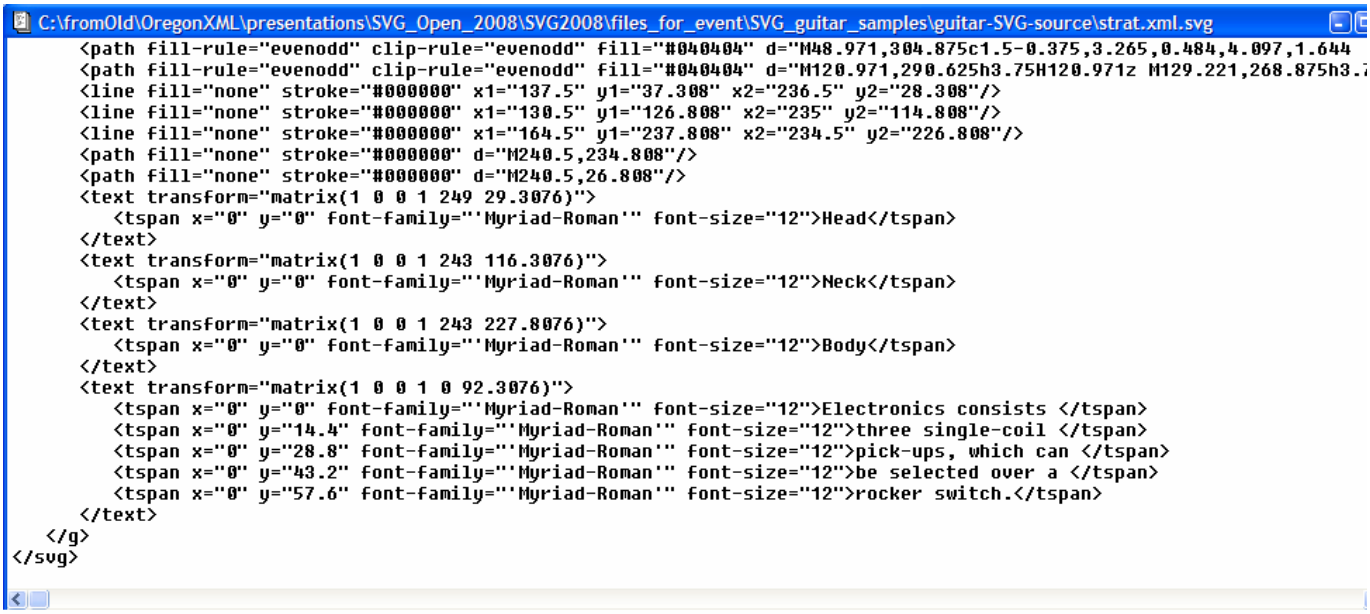
The following SVG file renders in graphic tools, and in browsers.

Figure 1. SVG File Rendered in a Browser



And the same file can be viewed in a text editor as plain text.

Figure 2. SVG File Rendered in a Text Editor



XLIFF

XLIFF is an open standard for translation. It was developed by representatives from all aspects of the industry, including tool makers, LSPs, downstream users, and translation customers. XLIFF is easy for LSPs to process; it isolates the text strings into source and target translation units. XLIFF has standardized methods for automating workflow, translation word counts, translation memory and segmentation.

Additional Translation Standards

Several other open standards for translation are worth mentioning. LISA **TMX** (Translation Memory eXchange, `http://www.lisa.org/Translation-Memory-e.34.0.html`) is a powerful standard for Translation Memory. And ITS offers a set of elements and attributes that can be used with new DTDs/Schemas to support the internationalization and localization of documents. Also it provides best practice techniques for developers of DTDs/Schemas (showing them how to enable internationalization of their documents).

Additional Authoring Standards

Solving the classic problem of efficiently translating text in graphics when translating books, papers, articles, help systems, software, and web pages, presents a challenge faced by many. Some very good authoring open standards exist, that work nicely with SVG and XLIFF, and help to solve this problem. DITA provides a document creation and management specification that builds content reuse into the authoring process. Docbook provides a system for writing structured documents . XHTML is the XML standard for markup rendered in browsers for the Internet.

. . . and One Standard to Bind Them?

There is another emerging OASIS standard called **OAXAL** (Open Architecture for XML Authoring and Localization Reference Model). OAXAL represents a method to build on technical documentation assets

by extending the usefulness of core XML-related standards in a comprehensive Open Standards based architecture. The OAXAL allows system builders to create an integrated environment for document creation and localization, with reference to the primary localization and XML document standards.

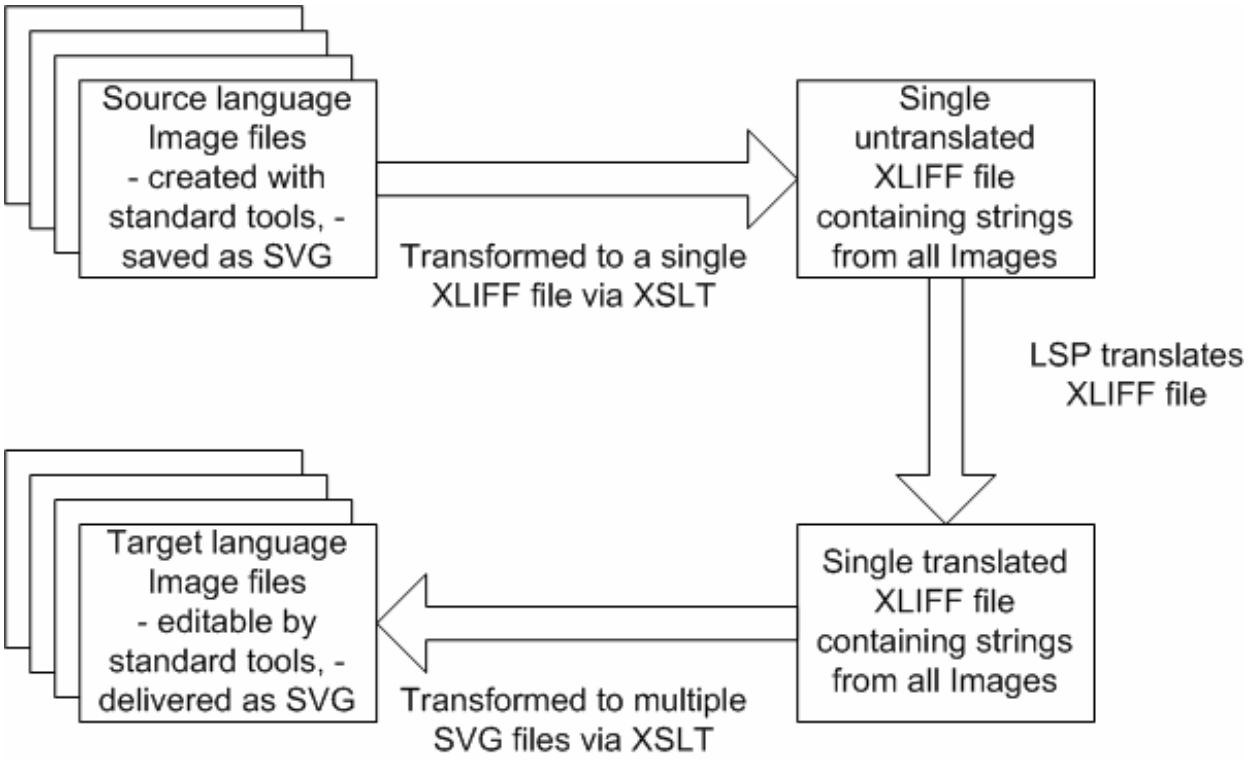
While this exciting young standard shows much promise, it is not mature enough for review in this paper.

How SVG and XLIFF Solve the Graphics Problem

The common denominator across the open standards that makes this approach work is XML.

- Since SVG files are XML, the strings that need to be translated are accessible via XSLT (eXtensible Stylesheet Language Transformation).
- Since XLIFF files are XML, the strings can be easily assembled via XSLT.

Figure 3. Basic SVG/XLIFF/SVG Process



- Many good commercial and open source tools exist that support SVG and XLIFF.
  - It is not difficult for graphic artists to create graphics with their tool of choice, then save the file as SVG.
  - Competent LSPs can easily process XLIFF files with their tool of choice. The format from which the XLIFF was derived has no impact.
- It is not difficult to create, or adapt processes to put in place a stable, automated workflow for translating SVG files via XLIFF.

Advantages of Open Standards

- Future-proof vs. proprietary:

Traditionally, technical communicators, or publishing groups have set their operation up around a toolset. It is not uncommon for groups to be casually identified as a "Framemaker house," or an Interleaf or Robohelp group. For graphics it is not uncommon to see a group standard set around Abode Illustrator, Corel Draw, AutoCad, or others. These are great tools, but there is a danger of locking your content assets into proprietary formats. With open standards, the tools can come and go, but the content assets (data and graphic files) are not at risk. The assets are future-proofed.

- Choice of tools (mix, match, change)

  Many good tools support XML for authoring, SVG for graphics, and XLIFF for translation. As stated previously, as long as these standards are used for authoring, archiving, and exchange, you are not married to a specific tool due to its proprietary format.

- Support of a large, noncommercial community

  An immense advantage to open standards is that they are not "owned" by any commercial entity. They are designed, maintained and supported by the public.

Advantages of Automation

- Automate your layout - no extra cost per each language

  In a non-automated process, each graphic must be processed individually. The time it takes to process a group of graphic files for translation increases with the number of graphics. And it increases with the number of languages. Smartly automating this means that with a "single-push-of-a-button" the entire job can be processed at once.

- Automate the text string/graphic layout - no errors

  An important aspect of automating to an open standard is that each case is predictable, the format is constant, and smart automation ensures that human errors can be eliminated.

  In a non-automated process the human is capable of, for example, accidentally not selecting an entire string to copy. Then when the string is pasted into the target graphic file, the person pasting might not spot the truncated string. It is likely that the human could increase the chances of not spotting the error if they do not understand the language. Similarly, spell-check is not likely to catch the error.

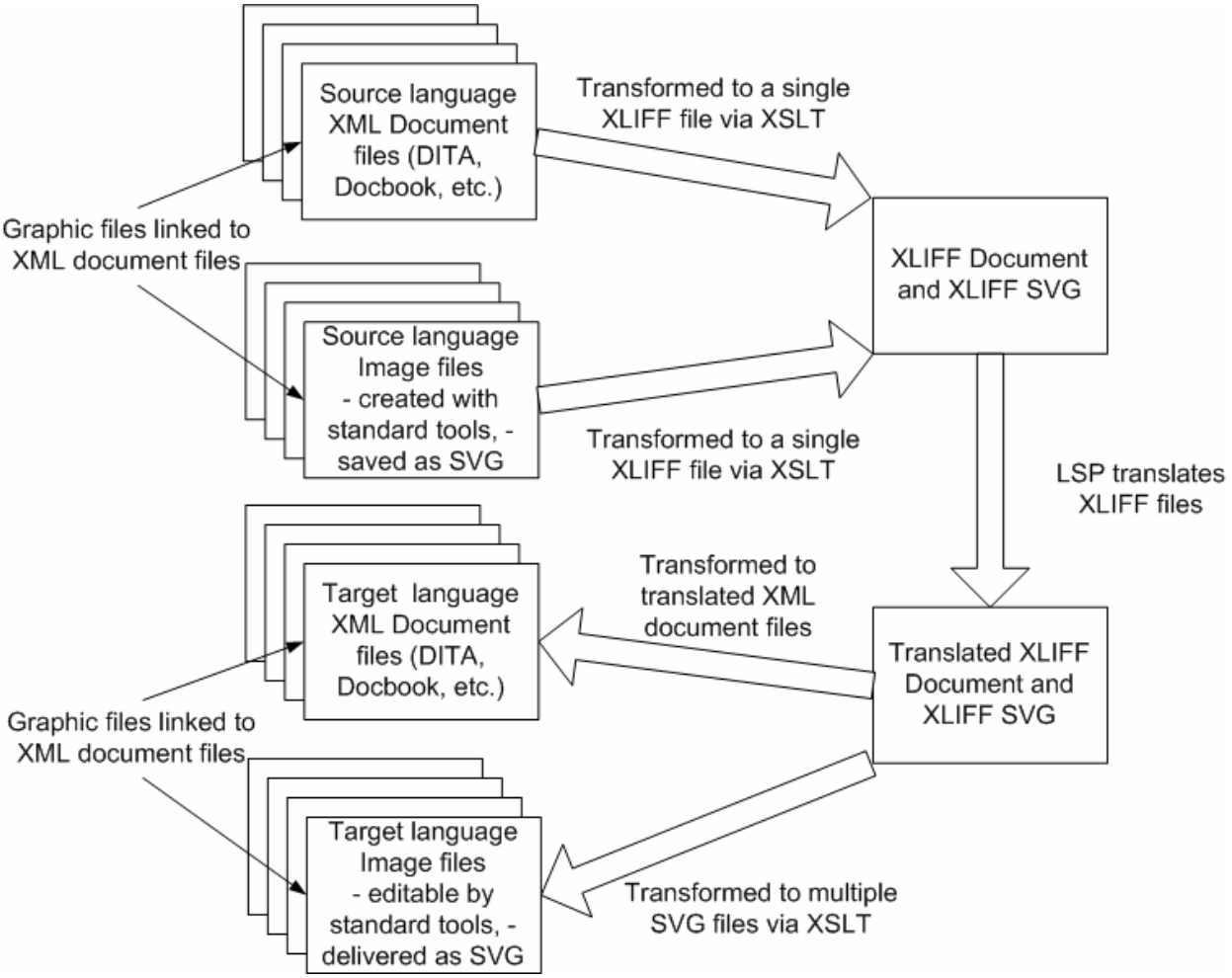
  In an automated process, the location of strings in an SVG file are prescribed by the standard. The strings are programmatically processed. The processing of the strings in the XLIFF files are similarly prescribed by the XLIFF standard. As long as the program is written in compliance with the two standards, human errors are eliminated.

  The advantage of open standards over proprietary solutions is that the open standards are not as vulnerable to not-predicted changes due to commercial factors.

- Leverage previous translations (not just of graphics)

Obviously, documentation groups use open standards for more than just graphics. XML open standards are commonly used for documents as well. Standards like Docbook, DITA, and XHTML are preferable to proprietary solutions for the same reasons already stated.

Figure 4. Translating XML Documents and Graphics with XLIFF



The localization industry has a tool called TM (translation memory) that allows translators to leverage previous translations. The open standards body LISA has an open standard for TM called TMX. Smart translation tools can process TMX so that a customer's valuable TM asset is not at risk, which it can be when in a proprietary TM format.

By processing the SVG files as XLIFF, then comparing the strings to previously translated strings in TM, the previous translation can be leveraged, which reduces cost and time.

Advantages of Not Requiring Extra Skill Sets from Your Translator

Think of this in terms of having a house painter install a new door for you. While some painters might be good at it, their core competency is painting (hopefully). This analogy can be applied to the translation industry. The person who creates the graphic art is good at using graphic art tools. The translator is good at translating.

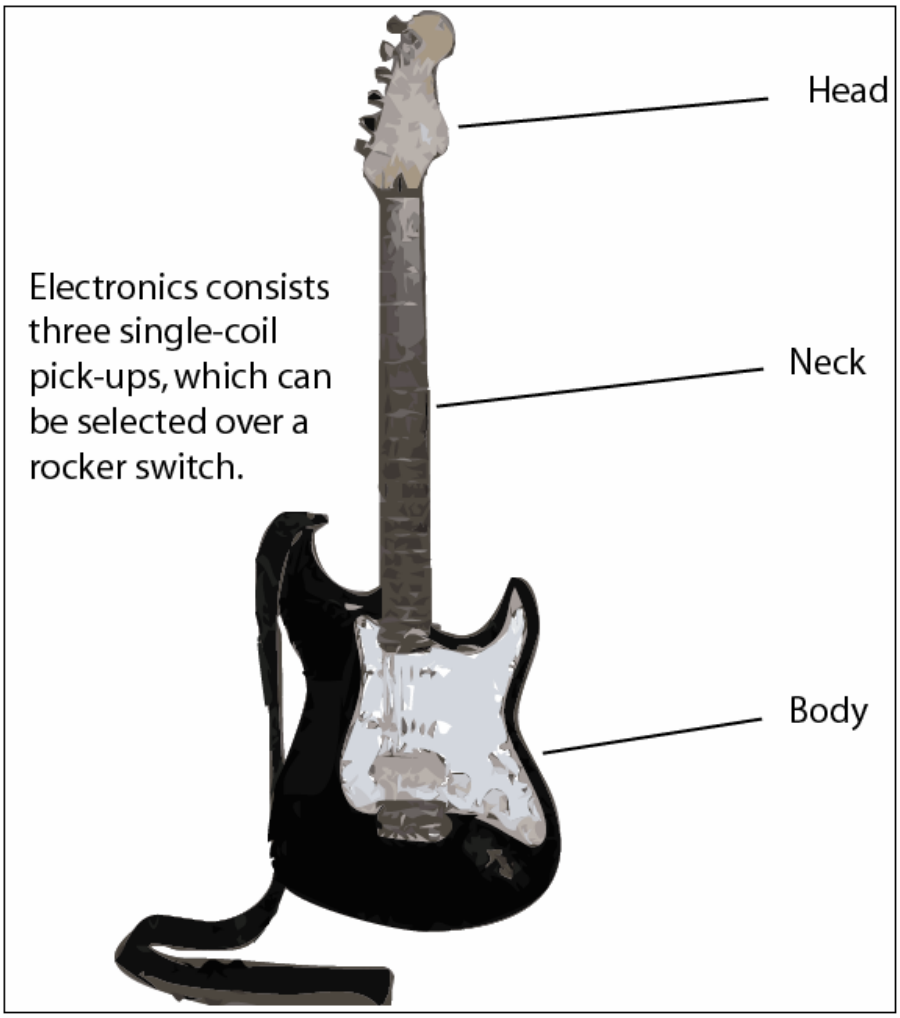
Example Architecture of Translating SVG files via XLIFF

The following example will illustrate how to:

1. Start with the SVG picture or the guitar previously shown
2. Extract the text and XML hierarchy in a source XLIFF file
3. Run the XLIFF file against the TMX file to look for previously translated strings
4. Transform the translated XLIFF file back to SVG

Start with an SVG file with English strings:

Figure 5. English SVG File



The text strings are accessible as `<tspan>` elements:

```
<text transform="matrix(1 0 0 1 249 29.3076)">
<tspan x="0" y="0" font-family="'Myriad-Roman'" font-size="12">Head</tspan></text>
<text transform="matrix(1 0 0 1 243 116.3076)">
<tspan x="0" y="0" font-family="'Myriad-Roman'" font-size="12">Neck</tspan></text>
<text transform="matrix(1 0 0 1 243 227.8076)">
<tspan x="0" y="0" font-family="'Myriad-Roman'" font-size="12">Body</tspan></text>
```

```
<text transform="matrix(1 0 0 1 0 92.3076)">
<tspan x="0" y="0" font-family="'Myriad-Roman'" font-size="12">Electronics consists
</tspan>
<tspan x="0" y="14.4" font-family="'Myriad-Roman'" font-size="12">three single-coil
</tspan>
<tspan x="0" y="28.8" font-family="'Myriad-Roman'" font-size="12">pick-ups, which can
</tspan>
<tspan x="0" y="43.2" font-family="'Myriad-Roman'" font-size="12">be selected over a
</tspan>
<tspan x="0" y="57.6" font-family="'Myriad-Roman'" font-size="12">rocker
switch.</tspan>
</text>
```

Extract the text strings into XLIFF `<trans-unit>` elements:

```
<trans-unit id="d0e445" restype="x-tspan" xmrk:locate_id="0">
<source xml:lang="en">Head</source>
<target state="needs-translation" xml:lang="DE">Head</target></trans-unit>
<trans-unit id="d0e451" restype="x-tspan" xmrk:locate_id="1">
<source xml:lang="en">Neck</source>
<target state="needs-translation" xml:lang="DE">Neck</target></trans-unit>
<trans-unit id="d0e457" restype="x-tspan" xmrk:locate_id="2">
<source xml:lang="en">Body</source>
<target state="needs-translation" xml:lang="DE">Body</target></trans-unit>
<trans-unit id="d0e463" restype="x-tspan" xmrk:locate_id="3">
<source xml:lang="en">Electronics consists </source>
<target state="needs-translation" xml:lang="DE">Electronics consists
</target></trans-unit>
<trans-unit id="d0e466" restype="x-tspan" xmrk:locate_id="4">
<source xml:lang="en">three single-coil </source>
<target state="needs-translation" xml:lang="DE">three single-coil </target></trans-
unit>
<trans-unit id="d0e469" restype="x-tspan" xmrk:locate_id="5">
<source xml:lang="en">pick-ups, which can </source>
<target state="needs-translation" xml:lang="DE">pick-ups, which can </target></trans-
unit>
<trans-unit id="d0e472" restype="x-tspan" xmrk:locate_id="6">
<source xml:lang="en">be selected over a </source>
<target state="needs-translation" xml:lang="DE">be selected over a </target></trans-
unit>
<trans-unit id="d0e475" restype="x-tspan" xmrk:locate_id="7">
<source xml:lang="en">rocker switch.</source>
<target state="needs-translation" xml:lang="DE">rocker switch.</target></trans-unit>
```

Run the XLIFF file against the existing TMX file to flag and replace already translated strings.`<tu segtype="block" tuid="g7">`

```
<tuv xml:lang="en-US"> <seg>neck</seg> </tuv>
<tuv xml:lang="de-DE"> <seg>Hals</seg> </tuv> </tu>
<tu segtype="block" tuid="g8">
<tuv xml:lang="en-US"> <seg>head</seg> </tuv>
<tuv xml:lang="de-DE"> <seg>Kopf</seg> </tuv> </tu>
<tu segtype="block" tuid="g9">
<tuv xml:lang="en-US"> <seg>body</seg> </tuv>
<tuv xml:lang="de-DE"> <seg>Korpus</seg> </tuv> </tu>
<tu segtype="sentence" tuid="g22">
<tuv xml:lang="en-US"> <seg>Electronics consists three single-coil pick-ups, which
can be selected over a rocker switch.</seg> </tuv>
```

```
<tuv xml:lang="de-DE"> <seg>Die Elektronik besteht aus drei Single-Coil-Tonabnehmern,
die Ã¼ber einen Kippschalter angewÃ¤hlt werden kÃ¶nnen</seg> </tuv> </tu>
```

Then set the state attribute to “needs-review-translation.”

Create the translated XLIFF file:

```
<trans-unit id="d0e445" restype="x-tspan" xmrk:locate_id="0">
<source xml:lang="en">Head</source>
<target state="needs-review-translation" xml:lang="DE">Kopf</target></trans-unit>
<trans-unit id="d0e451" restype="x-tspan" xmrk:locate_id="1">
<source xml:lang="en">Neck</source>
<target state="needs-review-translation" xml:lang="DE">Hals</target></trans-unit>
<trans-unit id="d0e457" restype="x-tspan" xmrk:locate_id="2">
<source xml:lang="en">Body</source>
<target state="needs-review-translation" xml:lang="DE">Korpus</target></trans-unit>
<trans-unit id="d0e463" restype="x-tspan" xmrk:locate_id="3">
<source xml:lang="en">Electronics consists </source>
<target state="needs-review-translation" xml:lang="DE">Die Elektronik besteht aus
</target></trans-unit>
<trans-unit id="d0e466" restype="x-tspan" xmrk:locate_id="4">
<source xml:lang="en">three single-coil </source>
<target state="needs-review-translation" xml:lang="DE">drei Single-Coil-Tonabnehmern,
</target></trans-unit>
<trans-unit id="d0e469" restype="x-tspan" xmrk:locate_id="5">
<source xml:lang="en">pick-ups, which can </source>
<target state="needs-review-translation" xml:lang="DE">die Ã¼ber einen
</target></trans-unit>
<trans-unit id="d0e472" restype="x-tspan" xmrk:locate_id="6">
<source xml:lang="en">be selected over a </source>
<target state="needs-review-translation" xml:lang="DE">Kippschalter angewÃ¤hlt
</target></trans-unit>
<trans-unit id="d0e475" restype="x-tspan" xmrk:locate_id="7">
<source xml:lang="en">rocker switch.</source>
<target state="needs-review-translation" xml:lang="DE">werden
kÃ¶nnen.</target></trans-unit>
```

Then transform the translated XLIFF file to a translated SVG file:

Figure 6. German SVG File

A Vision: Embedded SVG Markup in DITA, Translated via XLIFF

My vision is that we will be able to include SVG namespaced markup in DITA topics. From there we could send the entire translation job, multiple DITA Topics and Maps - with multiple SVG images, to the LSP as a single XLIFF file. The LSP could then translate the XLIFF file. The translated XLIFF file could then be transformed back into a translated, multi-topic, multi-mapped DITA project, rich with translated images.

Summary

For some of us, translation is an exciting task. For most of us, translation is complex, challenging, and usually difficult. Nearly universally, translating text in graphics is the most challenging and difficult aspect. As the need to translate increased, many very good tools, translation companies, and standards evolved to make the challenge achievable.

One very compelling method is to ensure stability, efficiency, and a future-proof translation process by enabling your processes, tools and translators with open standards designed for the task. When it comes to translating graphics, using SVG and XLIFF results in savings in time and cost. Leveraging this architecture with other open standards increases the benefit. Graphics are often part of an overall documentation publishing process. Open standards for documentation, like DITA, Docbook, and XHTML fit this architecture. They are compatible with SVG graphics, and are easily incorporated into a translation process that uses XLIFF at its core. And open standards for enhancing translation, like TMX, to leverage translation memory, also enhance the process.