revista.tradumàtica
tecnologies de la traducció

74-83

Bringing industry standards to Open Source localisers: a case study of Virtaal

Lucía Morado Vázquez
Centre for Next Generation Localisation
Localisation Research Centre
University of Limerick
lucia.morado@ul.ie

Friedel Wolff
Translate.org.za
The African Network for Localization
friedel@translate.org.za

ABSTRACT

The XML Localisation Interchange File Format (XLIFF) is an open standard promising interoperability and tool independence. It might be thought of as a natural fit for Open Source localisation, yet the Gettext PO format remains the de facto standard in Open Source localisation. We present a case study of the XLIFF implementation in Virtaal – an Open Source localisation tool supporting multiple formats. The primary target user group of Virtaal is made up of localisers of Open Source software – often volunteers. We study the implementation choices adopted by the developers with specific focus on the workflow metadata in XLIFF. In this regard we propose recommendations for simplification that hopefully improve XLIFF for use by a wider audience in future.

Keywords: CAT tools, XLIFF, FOSS, Virtaal, support, standards

RESUM *(Els estàndards de la indústria en la localització de codi obert: el cas de Virtaall)*

El format XLIFF (XML Localisation Interchange Format) és un estàndard obert que vol facilitar la interoperabilitat en localització així com la independència d'eines específiques. Es podria considerar un format ideal per a la localització de programari de codi obert, però el format PO de Gettext continua sent l'estàndard de facto en aquest tipus d'entorns. En aquest article presentem un estudi de cas sobre la implementació de XLIFF en Virtaal, una eina per a la localització de programari de codi obert compatible amb diversos formats. Virtaal es dirigeix principalment a localitzadors de programari obert, que sovint són voluntaris. Hem estudiat les solucions que els desenvolupadors van adoptar durant la implementació, especialment en relació a les metadades relatives al flux de treball en XLIFF. En aquest sentit, proposem algunes recomanacions per simplificar aquest estàndard que esperem que puguin contribuir a millorar XLIFF i que pugui ser utilitzat per un major grup d'usuaris en el futur.

Paraules clau: eines TAO, XLIFF, FOSS, Virtaal, suport, estàndards

RESUMEN *(Los estándares de la industria en la localización de código libre: el caso de Virtaal)*

El formato XLIFF (XML Localisation Interchange Format) es un estándar abierto pensado para facilitar la interoperabilidad y la independencia respecto a herramientas. A pesar de que XLIFF puede parecer una solución ideal para la localización de software de código abierto, el formato PO de gettext continúa siendo el estándar de facto en este tipo de entornos. En este artículo presentamos un estudio de caso sobre la implementación de

XLIFF en Virtaal, una herramienta para la localización de software de código abierto compatible con varios formatos. Virtaal se dirige principalmente a los localizadores de software abierto, muchos de ellos voluntarios. Hemos estudiado las soluciones adoptadas por los desarrolladores durante su implementación, en especial las relacionadas con los metadatos de flujo de trabajo en XLIFF. En este sentido, hacemos algunas propuestas para simplificar este estándar que esperamos puedan contribuir a mejorar XLIFF y a ampliar su círculo de usuarios en el futuro.

Palabras clave: herramientas TAO, XLIFF, FOSS, Virtaal, soporte, estándares

Introduction

Open standards are a natural fit for Free and Open Source Software (FOSS). They provide a universal representation of data and processes that can be interpreted and implemented without the risk of losing independence and accessibility. In the localisation process, due to its nature, it is necessary to deal with numerous formats, many of which are being developed every day or are constantly evolving. CAT (Computer Aided Translation) tools, consequently, have to interpret those formats and try to deal with them in order to facilitate the work of the translator or localiser (normally, by extracting the translatable text and protecting the rest of the content).

While the commercial localisation industry handles the issue of multiple formats with the development and adoption of XLIFF, FOSS localisation is dominated by the de facto standard of Gettext PO used in the majority of FOSS localisation projects (Frimannsson and Hogan 2005:10).

XLIFF is currently maintained by OASIS (Organization for the Advancement of Structured Information Standards). The standard maintenance and development is carried out by the XLIFF Technical Committe, which is composed by industry experts, CAT tool developers and localisation academics who meet twice a month. The current version is 1.2, and a new one (XLIFF 2.0) is under development and discussion. The concept behind XLIFF is extracting “the source localization-related data from the original format, and merging it back in place after the localization has been done.” (XLIFF TC 2007). An XLIFF document can contain several files, each divided into two parts: the header, in which the metadata of the document can be specified; and the body, in which the extracted localisable information is divided into translation units. Each translation unit consists of a source and a target element. An alternative translation can also be represented, for example translation suggestions from translation memory systems or machine translation. By definition, an XLIFF document is bilingual, however, using the xml:lang mechanism other languages can be also represented.

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
 <file original="Hello.txt" source-language="en" target-language="es" datatype="plaintext">
  <body>
   <trans-unit id="1" translate="yes" approved="yes">
    <source>Hello, Tradumàtica!</source>
    <target state="signed-off">¡Hola, Tradumàtica!</target>
   </trans-unit>
   <trans-unit id="2" translate="yes" approved="yes">
    <source>Let me introduce you to Virtaal</source>
    <target state="signed-off">Te presento a Virtaal</target>
   </trans-unit>
   <trans-unit id="3" translate="yes">
    <note from="programmer">Please translate this informally</note>
    <source state="signed-off">Hello, <ph>%s</ph>! How are you?</source>
    <target>Hola, <ph>%s</ph>! ¿Qué tal?</target>
    <alt-trans origin="Jose">
        <source>Good morning, <ph>%s</ph>! How are you?</source>
        <target>Buenos días, <ph>%s</ph>! ¿Qué tal?</target>
    </alt-trans>
   </trans-unit>
   <trans-unit id="4" translate="yes">
    <source>It uses a simple column layout</source>
    <target></target>
   </trans-unit>
   <trans-unit id="5" translate="yes">
    <source>Please translate me!</source>
    <target></target>
   </trans-unit>
   <trans-unit id="6" translate="yes">
    <source>Please translate me too!</source>
    <target></target>
   </trans-unit>
  </body>
 </file>
</xliff>
```

Figure 1. An example XLIFF file

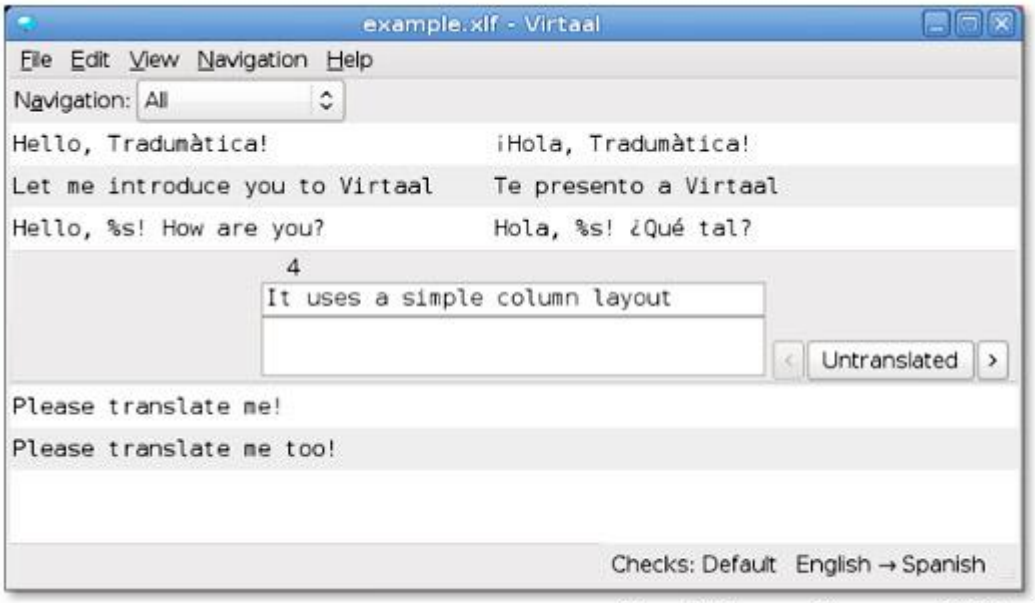

Figure 2. The same file processed in Virtaal

Although the PO format is still the prevailing interchange format in the localisation of FOSS, the XLIFF format has also been adopted by many FOSS CAT tools (Cánovas and Samson 2008 pp.46-52 and Diaz 2008 pp. 60-70). In a presentation given in the 1st XLIFF Symposium, Frimannsson and Lieske claimed that the support for XLIFF in FOSS CAT tools was poor with "the notable exception of Virtaal" (2010:4). Frimannnsson and Hogan (2005) worked on the adoption of the XLIFF format by the OS community. They conducted research on both formats (PO and XLIFF) and defined an *XLIFF 1.2 Representation Guide for Gettext PO* that was later presented and approved by the XLIFF Technical Commitee (XLIFF TC 2006). The advantages of XLIFF over PO, according to Frimannsson and Hogan (2005:14-15), are: allowance of the insertion of more metadata items; more specific workflow information; better possibilities of sharing translation memories between different projects; decoupling of localisation technologies; and XML-based processing and the localisation of non-textual metadata. Although it is a fascinating topic, the discussion of which format should be adopted in FOSS is beyond the scope of this paper, instead we refer interested readers to the afore mentioned paper by Frimannsson and Hogan (2005) and Mata (2008).

The structure of this paper is as follows: in the next section, we present a background on Virtaal, the CAT tool considered in this case study. In the following sections we present a high level view of XLIFF support in CAT tools in general, and in Virtaal in particular. Then follows a more detailed discussion of workflow metadata in XLIFF, with details about the implementation thereof in Virtaal. The paper concludes with a description of the underlying workflow model in Virtaal, which we propose as a modification for the upcoming version (2.0) of the XLIFF standard.

1. Virtaal and Translate.org.za

Virtaal was developed by Translate.org.za, a South African localisation company working on the support for South African languages in software. An overview of its localisation activities were given by Wolff (2006). The initial development of localisation tools was necessary to address some shortcomings in the Free and Open Source Software (FOSS) tools that were available at the time. These initial tools, collected as part of the Translate Toolkit[1] soon attracted the interest of localisers all over the world, especially for smaller language communities. The development of localisation tools for a wider audience gradually became a bigger part of the work at Translate.org.za. The tools now form part of the workflow of several localisation teams for major FOSS projects, such as OpenOffice.org, LibreOffice, Firefox, Thunderbird and others.

The development of a CAT tool was not an initial goal – several applications were available, although most were limited to the Gettext PO format. Following the work of Frimannsson (2005) there was interest in better support for XLIFF but initial tool support was very limited with very few of the existing tools being extended to support XLIFF to any meaningful level. Another shortcoming observed at training events was the learning curve required for some applications, especially to obtain all functionality such as translation memory. Also, most of the FOSS CAT tools were only available on Linux and other UNIX type systems.

These were some of the deciding factors for the development of a new CAT tool, Virtaal. It runs on several operating systems and supports editing of several file formats, including Gettext PO, XLIFF, TMX and TBX. Its main aim is to allow anyone, regardless of experience, to translate productively without sacrificing quality[2]. The focus is therefore foremost on simplicity but it still provides many of the features common in CAT tools, often in simpler

1 For more details, visit the website: http://translate.sourceforge.net/wiki/toolkit/index
2 For more details, visit the website: http://translate.sourceforge.net/wiki/virtaal/index

forms, such as translation memory that needs no configuration. It has a simple layout, emphasising context in the file above functional features of the CAT tool.

2. XLIFF support in CAT tools

In the localisation standards field, a topic that is frequently under discussion is the need for standard compliance mechanisms that allow users to test and certify the grade of support and implementation that certain CAT tools provide for specific standards.

One main aspect of this discussion is whether this standard compliance should be studied in the performance of the CAT tool or in the document produced or modified by the tool (XLIFF TC 2010). In the specific case of the XLIFF standard the point has been raised that, since XLIFF was a document-based standard, the analysis should be conducted only on the document produced (XLIFF TC 2010 and Raya 2011a).

In related fields other standards bodies like the W3C have provided the community with mechanisms to test standard compliance at the document level. For example the W3C Markup Validation Service "checks the markup validity of Web documents HTML, XHTML, SMIL, MathML, etc" (W3C 2011). At the implementation level, the Web Standards Project has developed, amongst others, the ACID2 Browser Test "written to help browser vendors make sure their products correctly support features that web designers would like to use." (Web Standard Project 2011b). Validity at the document level in XLIFF can be tested with the tool XLIFF Checker, developed by Rodolfo Raya under the Eclipse Public License (Raya 2011b). XLIFF checker only analyses the XLIFF documents produced or modified by another tool, not the behaviour of the tool itself.

Even though the TC doesn't certify compliance of CAT tools to the standard, there have been independent studies carried out to analyse the XLIFF standard compliance of CAT Tools. We analyse in the following paragraphs two studies carried out in 2010: the first one carried out by Thomas Imhof and the second one by Micah Bly.

The study by Thomas Imhof divided XLIFF support in various CAT tools into three distinct levels. Tools on level 1 have the lowest support where "XLIFF elements were not treated at all" (Imhof 2010:21). On level 2, at the very least, the source and target elements were interpreted properly (Imhof 2010:22). And on level 3, the tools had the highest XLIFF support and interpreted "most or all standard XLIFF elements" (Imhof 2010:23). Imhof's study consisted of 10 CAT tools: five of the tools were classified as level 1 grade, two of them as level 2 and three were classified as level 3. Only one of the analysed CAT tools was FOSS.

In September 2010 a broader study was presented by Micah Bly. He analysed 17 CAT tools (four of which were FOSS). The tools were divided into two categories: XLIFF generators and XLIFF editors. "Generators**"** included "any tool that creates any XLIFF from another file(s), whether it is a server-based system, or a desktop-tool." (Bly 2010:1), 13 CAT tools were labelled as "generators". The "editors" category included "any tool that can open an XLIFF file, allow you to edit it, and then save it in XLIFF format" (Bly 2010:4). 9 CAT tools were included in this category, four of them are also classified as generators. In this classification, Virtaal should be included in the latter category, as XLIFF editor.

Bly's study recorded the support of XLIFF elements in the analysed CAT tools as illustrated in the tables 1 and 2. His analysis also included a study of some CAT tool features that fall outside the scope of this paper.

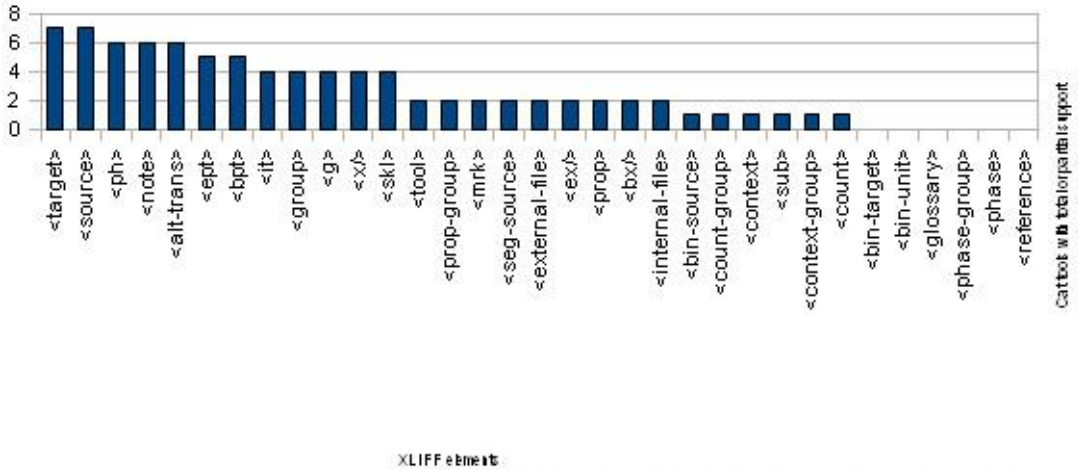

Table 1. Analysis of XLIFF Editors, adapted from Bly 2010

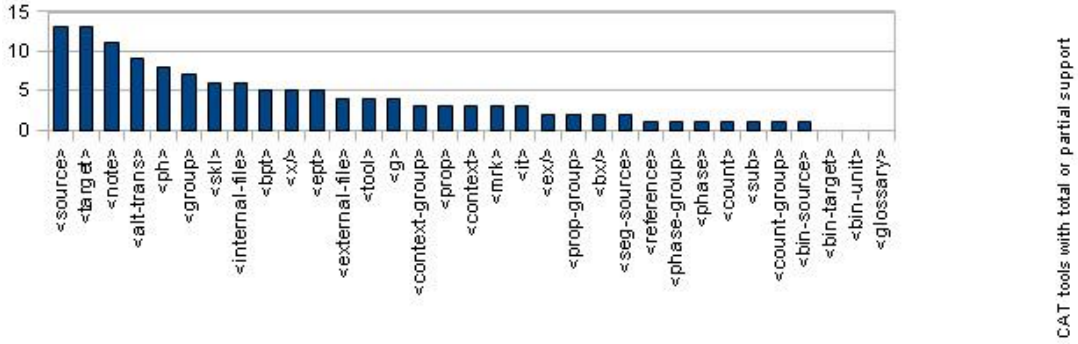

Table 2. Analysis of XLIFF Generators, adapted from Bly 2010

Bly's study raised interest when presented at the 1st XLIFF International Symposium and the XLIFF TC (Lieske 2010). Conversations between Bly and the TC followed and in March 2011 he ”approve[d] maintenance of the matrix to be taken over by the TC“ (XLIFF TC 2011). The work of this paper will be added to that effort and will be made publicly available through the XLIFF TC Wiki.

3. XLIFF Implementation in Virtaal

Building on Bly's analysis, we have analysed Virtaal's support for XLIFF elements. We found support for 14 of the XLIFF elements contained in Bly's analysis, well above the average number of supported elements achieved by other tools in Bly's original study.

As we can see from this summary and Bly’s data, several elements were not widely implemented leading to the conclusion that support for them is not that important or useful for interchange since a recipient's software is not likely to support it. A shortcoming with Bly's analysis was that it carried no sense of the relative importance of different parts of the XLIFF specification. To measure the level of support for XLIFF software, a “weighted sum model”

would be useful in order to give a larger weight for the commonly supported features in XLIFF that represent good interchange ability between XLIFF software. It is also notable that support is measured only in terms of elements and does not include attributes and their values.

The interoperability of tools with regard to the elements' attributes and their values is important for interchange, especially with regards to workflow information. We discuss this part of Virtaal's XLIFF support in the next section.

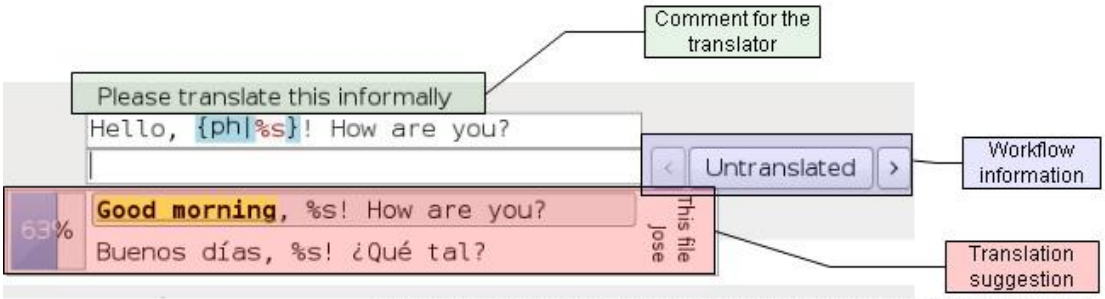

Figure 3. A detailed view depicting support for several XLIFF elements

4. Workflow information in XLIFF and Virtaal

There are three mechanisms in XLIFF that incorporate workflow information. The first one is the <phase-group> element, in which the user can specify the phase of the process that the document is currently in. Virtaal does not support this element; this is common to all of the XLIFF editors in Bly's study.

The second mechanism is the “approved” attribute that can be included in the <trans-unit> or <bin-unit> elements. This attribute has a boolean value: “yes” if the translation has passed its last review and is in its final state or “no” if it has not achieved its final state. Virtaal supports this feature, and adds the “approved” attribute with a “yes” value once the translator marks a translation as “translated” or “reviewed”. If the translator changes the state to indicate “Needs Work” or “Needs Review”, the tool internally changes the value “yes” to “no”.

The third method for adding workflow information to an XLIFF document is through the “state” attribute in the elements <target> or <bin-target>. It has ten predefined values, they are: “final”, “needs-adaptation”, “needs-l10n”, “needs-review-adaptation”, “needs-review-l10n”, “needs-review-translation”, “needs-translation”, “new”, “signed-off” and “translated”. Virtaal supports this mechanism and has simplified it to its own needs.

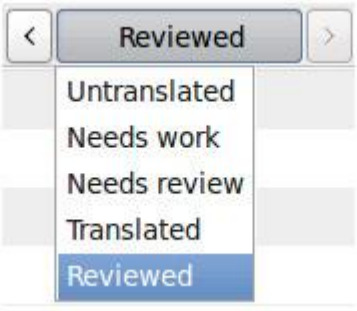

Figure 4. A detailed view of the supported workflow states in Virtaal

The complexity of the state attribute in particular is one of the most quoted criticisms of the current XLIFF specification (Morado *et al* 2010, Savourel 2010 and Wolff 2010). Virtaal's philosophy of simplifying things and making the translator's job easier led to the condensing of the 10 predefined values of the state attributes by mapping the ten values to a subset of five (Untranslated, Needs Work, Needs Review, Translated and Reviewed). See the implementation diagram in figure 5 to see how they are matched with the predefined values of the XLIFF state attribute.

It is important to note that if there is not information in the target element but the value of the attribute indicates that it is translated or in any other state, the tool will display that state (in other words, the tool will follow the value to be displayed in the workflow window). If however, there is some data inside the target element but the value attribute is set to “new”, the tool will show the state “Needs Work”.

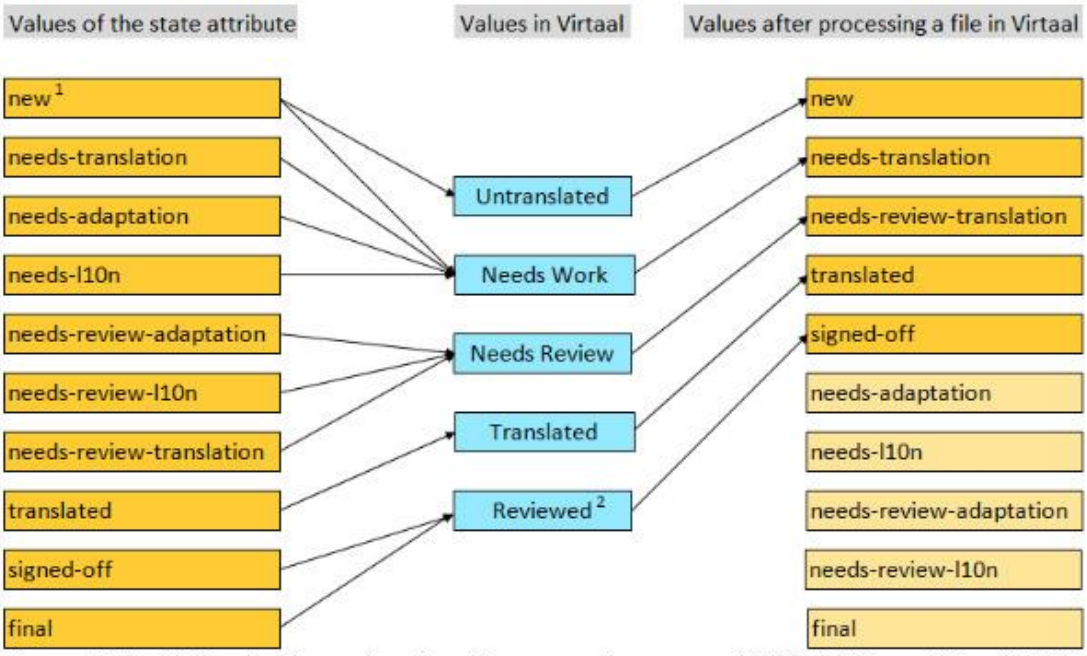

Figure 5. The bi-directional mapping of workflow states between an XLIFF 1.2 file and Virtaal's GUI

5. Proposal

The 'approved' and 'state' attributes seem to relay very similar pieces of information and are presented as a single piece of information in Virtaal. It is unclear what a combination of “approved=’no’” and “state=’final’” would mean, for example, therefore we suggest that such incompatible workflow information be avoided by combining it into a single field allowing only one of the allowable states.

The five states supported by Virtaal on their own should support reasonably complex workflows. The distinction between “needs-translation”, “needs-adaption” and “needs-l10n” is best handled as auxiliary data in an extra attribute (like state-qualifier from XLIFF 1.2) rather than as separate values in a state field. If the standard is adapted in this way, it would be important to clearly indicate which values of the state attribute are supposed to be augmented with each of the possible values of the state-qualifier. This would be important in avoiding uncertainty for tool developers about when to clear the state-qualifier in order to avoid outputting a file with a state-qualifier augmenting an unrelated state attribute.

Value	Description
needs-work	Indicates that the work is incomplete and needs further attention from the translator before it can be considered for review or consumption.
needs-review	Indicates that the unit is translated but needs review before it can be considered to be approval.
translated	Indicates that the unit is translated and, in the absence of a review phase, could be seen as approved.
reviewed	Indicates that the unit has been reviewed and is approved.

Table 3. State proposal.

Note that we are not proposing any value for an 'empty' or 'new' state. We have chosen not to include that in our proposal for two reasons: firstly, to avoid any inconsistencies, e.g. a target element with a state marked as “empty”, but containing data; and secondly, to allow for a simpler set of states, we consider that that the empty value can be removed and any relevant information identified and gathered by CAT tools based on either the absence of the target element or by the absence of content in its interior. A completely new and untranslated unit can be represented with state=”needs-work” and an empty target element.

This approach will be formally presented in the XLIFF TC for discussion and possible acceptance in the new version 2.0. It could also be integrated in the Service Oriented Localisation Architecture Solution (SOLAS) through the LocConnect orchestration framework described in this same volume by Wasala et al (2011).

Conclusion

In this paper we introduced an XLIFF editor, Virtaal, with relatively good XLIFF support compared to many alternative CAT tools, even commercial ones. Being born in the FOSS world, it has a strong focus on volunteers and part time translators. Its model for workflow is simpler than that of the current XLIFF 1.2 specification. It combines two attributes for workflow, namely “approved” and “state” and presents thems as a single piece of information for the user. In our view this simplified model is an improvement over the complexity of the workflow model in XLIFF 1.2, and we recommend that this approach be incorporated into future versions of XLIFF. Such an amendment will not result in a reduction in the level of detail supported but will simplify XLIFF to the benefit of tool developers and users.

References

Bly, M. (2010). “XLIFF Compatibility Chart”, in 1st XLIFF International Symposium. 22 September 2010, Limerick, Ireland. Available at <http://www.localisation.ie/xliff/resources/presentations/xliff_tools_matrix_20100922.numbers.pdf>.

Cánovas, M. and Samson, R., (2008). “Herramientas libres para la traducción en entornos MS Windows”, in Díaz, O. and García, M. (ed.); *Traducir (con) software libre,* Granada: Editorial Comares. 33-55.

Díaz, O. (2008). “Ferramentas livres para traduzir com GNU/Linux e Mac OS X”, in Díaz, O. and García, M. (ed.); *Traducir (con) software libre,* Granada: Editorial Comares. 57-73.

Frimannsson, A. and Hogan, J. (2005) “Adopting Standards-based XML File Formats in Open Source Localisation”, Localisation Focus 4(4): 9-23.

Frimannsson, A. and Lieske, C. (2010) “Next Generation XLIFF”, in 1st XLIFF International Symposium. 22 September 2010, Limerick, Ireland. Available at <http://bit.ly/tyIukd>.

Imhof, T. (2010). "XLIFF - a bilingual interchange format", in MemoQfest, 5-7 May 2010, Budapest, Hungary.

Mata, M. (2008). “Formatos libres en traducción y localización”, in Díaz, O. and García, M. (ed.); *Traducir (con) software libre,* Granada: Editorial Comares. 75-122.

Morado , L., Anastasiou, D., Exton, C., “XLIFF Interoperability Challenges”. Poster in CNGL Scientific Committee Meeting, 27-29 April 2010, Limerick, Ireland. Available at <http://bit.ly/bFt8Zt>.

Lieske, C. (2010). The XLIFF TC's Summary of the First XLIFF Symposium. <http://bit.ly/uyvW80>.Last updated: 08.10.2011. Page consulted on date: 10.04.2011

Raya, R. (2011a). Re: [xliff] XLIFF 2.O Core. <http://lists.oasis-open.org/archives/xliff/201104/msg00030.html>. In Internal XLIFF TC mailing discussions. Sent on: 08.04.2011. Page consulted on date: 10:04:2011.

Raya, R. (2011b). XLIFF Checker. Available at <http://www.maxprograms.com/products/xliffchecker.html>. Page consulted on date: 11:06:2011

Savourel, Y. (2010). “Translation State”, In OASIS XLIFF TC Wiki. Available at <http://wiki.oasis-open.org/xliff/XLIFF2.0/Feature/TranslationState>.

Wasala, A., O'Keeffe, I., Schäler, R. (2011). “Towards an Open Source Localisation Orchestration Framework”, in Tradumàtica núm. 9, localització i web.

Web Standard Project (2011a). Acid Tests <http://www.acidtests.org/>. Page consulted on date: 10.04.2011.

Web Standard Project (2011b). Acid2 Browser Test <http://www.webstandards.org/action/acid2/>. Page consulted on date: 10.04.2011.

Wolff, F. (2006). "Software Localisation by Translate.org.za", Localisation Focus 5(3): 19-21.

Wolff, F. (2010). "XLIFF from a volunteer's point of View", in 1st XLIFF International Symposium, 22 September 2010, Limerick, Ireland.

W3C. The W3C Validation Service <http://validator.w3.org/>. Page consulted on date: 10.04.2011.

XLIFF TC (2006). XLIFF 1.2 Representation Guide for Gettext PO. Available at: <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-po/xliff-profile-po-1.2-cd02.html>. Page consulted on date: 02.05.2011.

XLIFF TC (2007). XLIFF 1.2 White Paper. Page available at: <http://bit.ly/uifGKN>.

XLIFF TC (2010). Face to Face Limerick Minutes <http://lists.oasis-open.org/archives/xliff/201009/msg00026.html> Last updated: 21.09.2011. Page consulted on date: 10.04.2011

XLIFF TC (2011). XLIFF Teleconference - Mar-1-2011 Minutes <http://lists.oasis-open.org/archives/xliff/201103/msg00000.html> Last updated: 01.03.2011. Page consulted on date: 10.04.2011