Creación y mantenimiento de traducciones de software libre usando memorias de traducción

Gestión de traducciones usando memorias de traducción con gettext y Docbook/XML

Ismael Olea

Historial de revisiones
Revisión 0.0 Fecha sin cerrar
Versión de desarrollo inicial

Tabla de contenidos

Resumen

Se pretende explicar cómo aplicando herramientas libres existentes se pueden usar memorias de traducción con documentos Docbook-XML, lo cual ayuda a simplificar y acelerar el mantenimiento y sincronización de traducciones de documentos, lo cual ha sido hasta hoy necesariamente muy laborioso.

Estado actual del mantenimiento de los documentos en el mundo del software libre

```
- enumerar los formatos
- cómo se ha apañado la gente hasta ahora
- el caso de los programas
- el esfuerto de TLDP-ES y Juan Rafael
```

Partiendo de la idea de «Producto Acabado = Programa Software + Manual de Usuario» es obvio que las proyectos de aplicaciones libres acaben documentando a las mismas. Con el tiempo, en función de los orígenes históricos y el desarrollo de los proyectos y las tecnologías en el mercado se ha creado una pléyade de formatos paralelos que hacen todavía más complicado el poco simple esfuerzo de gestionar la creación y mantenimiento de documentación y sus traducciones.

Así es posible encontrarse desde texto más o menos estructurado, páginas del manual (man), ficheros PDF o PostScript, documentos compuestos con Tex/LaTeX o incluso con Troff, HTML (no siempre sintácticamente

correcto), el formato info/texinfo de GNU, Linuxdoc-SGML (incorrectamente llamado formato SGML en estos ambientes), Docbook (en las veriones SGML y XML) y en menor medida RTF, MSDOC, OpenOffice/StarOffice. También hay muchísimo material en la web en HTML más o menos integrado en «CMS» que hacen difícil gestionarlos integralmente con el resto de doc.

Afortunadamente son muchos los esfuerzos que están convergiendo en XML en general y en Docbook [http://www.oreilly.com/pub/a/oreilly/frank/oscon_summit.html] en particular.

Los que lo hacen a diario saben que trabajar con documentación es desgraciadamente muy poco trivial. Mantener sincronizada la traducción de un documento lo complica aún más. En el mundo profesional eso está básicamente resuelto con el tipo de herramientas que cubre este artículo, pero que se sepa ninguna puede ser ejecutiva de forma nativa en plataformas linux/*ix y tampoco son libres.

Así pues y hasta ahora, si el trabajo de traducir ha sido duro y artesano, el de mantener actualizadas las traducciones lo ha sido muchísimo más, siendo muchos los documentos traducidos abandonados y muy seriamente desactualizados. Y desde el momento en el que existe la técnica de Memorias de Traducción (en adelante MMTT), es un despilfarro inmenso de tiempo, recursos y de conocimiento, en la forma de los pares de traducción que podrían haberse almacenado y reutilizado posteriormente.

Qué es una memoria de traducción

Citamos a Corrigan y Foster

SUN

: [Translation memory systems, known as TM systems, work by looking up segments in a database containing a large number of previously translated segments and their translations. (Segments are pieces of source files, usually sentences, that can be translated reasonably independently.) The database might contain segments that match the input segment exactly or segments that are similar to the segment presented for translation. These translations are then provided to the translator as suggested translations for each segment.]

Osea, es una tabla con, básicamente, dos columnas que asocian un segmento (una frase, una expresión o una palabra) en un idioma con su traducción a otro.

¿Y cuál es la ventaja competitiva de las MMTT? Son varias. Al menos:

- Permite sistematizar la gestión y el mantenimiento de las traducciones con herramientas mecánicas.
- Facilita enormemente la actualización de las traducciones: al comparar una nueva versión del original con su memoria, se traducen automáticamente todas aquellas partes que no han variado, dejando visibles y directamente manejables los nuevos segmentos.
- Un par de traducción, el original y su traducción, son ciertos, o al menos lo son dentro del área de conocimiento del documento o al menos lo han sido en algún momento del ciclo de actualizaciones del mismo. Si el segmento original sigue vigente en el documento tiene sentido mantener su traducción. Si el segmento original desapareció, sería un desperdicio ignorar la vieja traducción siendo tan fácil archivarla con medios automáticos. No hay garantía ninguna pero tal vez en el futuro pudiera volver a ser una traducción necesaria o servir al menos como sugerencia para un nuevo segmento.

Y además podemos imaginar otras ventajas.

- Conforme se trabaja con total normalidad, se va creando un histórico del documento, lo que llegado el caso pudiera descubrirse como muy útil para algún fin específico.
- Si disponemos de un documento original con MMTT a varias lenguas, podemos tener indirectamente un mecanismo de traducción entre estas a través del idioma original.
- Colecciones de MMTT organizadas convenientemente pueden componer un corpus de traducciones de interés técnico o científico.
- Se pueden usar glosarios y terminologías de traducción como MMTT símplemente dándoles el formato adecuado y alimentando a nuestras herramientas de traducción. Esto puede significar un salto cualitativo de la calidad y homogeneidad de los trabajos de traducción.

Las MMTT, una técnica madura en el mundo profesional

```
Aplicaciones:
      - Trados
      - Wordtrans
      - Wordfast
      - DejaVù

Estándares:
      - TMX
      - XLIFF
```

Cómo se usa una MT

Vamos a mostrar el funcionamiento básico paso a paso.

Lo primero es crear la memoria:

1. Partimos de un documento original, en un idioma cualquiera.
2. Alimentamos a nuestra herramienta de MMTT (que podría ser la propia herramienta asistente para la traducción) con el documento original.
3. A partir de ahora obtenemos un fichero que es el documento original descompuesto en pares de traducción por cada segmento, obviamente. Según el algoritmo de descomposición que la herramienta tenga implementado estos segmentos podrán ser más o menos pequeños y por tanto la gestión de la MMTT será más fina y potente.
4. En este momento se realiza el trabajo de traducción, sin particularidad especial.
5. Una vez concluido el trabajo de traducción, se almacena una memoria de traducción del documento y se exporta la versión traducida del documento original en su mismo formato.

En este momento tenemos tres ficheros: el documento original, el documento traducido y la memoria de traducción del mismo, que sólo contiene los pares de traducción para cada segmento.

El siguiente proceso sería el de actualización de la traducción porque el documento original ha sido modificado. Pensemos por ejemplo que se trata de un versión corregida del mismo, o una versión nueva de una especificación técnica.

1. Tomamos la nueva versión del documento original y la memoria de traducción que creamos en la traducción original.

2. Alimentamos con ambos ficheros a nuestra herramienta de MMTT. De nuevo la herramienta descompondrá el documento en segmentos, y ahora buscará en la memoria segmentos traducidos iguales o suficientemente parecidos. Si encuentra uno igual y puede estar seguro de que se trata exactamente del mismo, incorporará el segmento traducido directamente. Si no tiene esa certeza o si el segmento no es idéntico, la herramienta tomará la traducción y la usará como una posible sugerencia.

3. A continuación empieza el trabajo del traductor. Ahora se encontrará que aparecerán como ya traducidos todos los segmentos que no se han modificado y como trabajo pendiente todos los segmentos para los que no se ha encontrado traducción y para aquellos para los que hay traducciones dudosas. Las traducciones dudosas se usan como sugerencias. El traductor puede elegir alguna de las opciones como la definitiva, utilizar alguna de ellas para escribir la definitiva o incluso ignorar todas ellas para escribir una traducción completamente diferente.

4. Una vez acabado el trabajo, se construye el documento traducido final, usando el formato original. Por otro lado, los nuevos segmentos traducidos (y se consideran traducidos porque el traductor los ha revisado y validado) se almacenan en la MT del documento.

5. En este momento tenemos varios ficheros: original v1, original v2, traducción v1, traducción v2 y una memoria de traducción de este documento que contiene los pares de traducción de todos los segmentos de las dos versiones. Nuevas revisiones del original servirán para alimentar de nuevo la memoria de traducción. Y si ahora está pensando si de la memoria se borra algo, la respuesta es no. En general la memoria siempre se alimenta con nuevas traducciones y no pierde información. Al fin y al cabo, si el traductor consideró que un segmento que ya no se usa era correcto en su momento ¿por qué no guardarlo si eso no supone ningún problema técnico?.

Y ahora ilustraremos el caso ideal de recuperación de segmentos. No hay ninguna garantía de que siempre ocurran casos así, pero ¿por qué desperdiciar la oportunidad si alguna vez surge sino causa ningún problema técnico?

Imaginemos que estamos trabajando en actualizar revisiones de especificaciones técnicas. En una de las versiones anteriores se propuso un borrador para una sección determinada que ha estado en discusión durante varias revisiones y alguna de ellas se retiró por cuestiones de implementación, por ejemplo. Como para cada una de esas revisiones se escribió una traducción, su memoria de traducción contiene el histórico de toda la vida del documento en forma de pares de segmentos traducidos.

Imaginemos ahora que en una de esas revisiones se considera que la tecnología ha cambiado y se recupera el viejo borrador como parte del documento. Al volver a procesar el original, en su más reciente versión, a través de la herramienta de gestión de la memoria de traducción, ésta encontrará las vieja traducción que se realizó varias versiones atrás y las recuperará automáticamente o al menos como sugerencias de traducción, que el traductor sólo tendrá que revisar rápidamente y aceptar. Cuanto más grande sea el texto, mayores serán el ahorro y la productividad.

Cómo se usa gettext para mantener las traducciones de programas

```
- Ídem, con un poco de detalle
- Los GUI
```

############################## VOY POR AQUÍ ##################################

Una alternativa libre

- presentar poxml
- cómo descargarlo, cómo encontrar las fuentes, cómo participar en el desarrollo y cómo contribuir reportes.
- explicar cada utilidad

Práctica: crear una MT nueva desde cero

- preparacion XML
- preparacion PO
- traducción
- reconstrucción

Práctica: actualizando una traducción usando MT

Práctica: creando una MT nueva a partir de traducciones anteriores

Problemas al trabajar con poxml

Sugerencias para el futuro desarrollo de poxml

- mantenerlo como un desarrollo independiente (como se hizo con scrollkeeper)
- documentarlo
- repaso de cariño a las aplicaciones.
- especificaciones.

Colofón

- Recomendación encarecida del uso de MMTT
- Realimentaciones sobre este trabajo:
    - mejoras en loor de la claridad
    - erratas o defectos
    - otras sugerencias
- Se buscan programatas para seguir trabajando en esta peli.

Bibliografía

- Artículos de JR
- el artículo del tío de Sun: http://developers.sun.com/dev/gadc/technicalpublications/articles/xliff.html

- 44
- Norma XML
- Norma TMX
- Norma XLIFF