

Localizer

Juan David Ibáñez Palomar Palomar

Copyright (C) 2002 Juan David Ibáñez Palomar, j-david@noos.fr. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. No habiendo Secciones Invariantes, ni Textos de Portada ni Textos de Contra Portada. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Localizer es una herramienta para la construcción de aplicaciones web en varios idiomas, trata tanto la internacionalización de las interfaces de usuario como la gestión de contenido multilingüe.

Cien por cien software libre, su desarrollo comenzó en una pequeña universidad española hace ya más de un año. Desde entonces ha seguido creciendo hasta alcanzar cierta popularidad.

Ahora se abre un nuevo período para este producto en el cual el objetivo es llenar un vacío en el mundo del software libre, dentro de la industria de la localización, tradicionalmente dominada por soluciones propietarias.

Tabla de contenidos

1. Introducción	1
1.1. Orígenes, el proyecto LLEU	2
1.2. Esto es software libre	2
1.3. Un producto Zope	2
1.4. Contenido del artículo	2
2. Internacionalización y localización	3
3. Productos	3
3.1. Gettext	3
4. Sitios web	4
4.1. El catálogo de mensajes	4
4.2. Contenido	5
4.3. Imágenes, fechas, etc..	5
5. Negociación de idioma	5
6. Unicode	6
7. Extensibilidad	6
8. Ejemplos	6
8.1. LLEU	7
8.2. EuroPython.org	7
9. Conclusiones	8
10. Referencias	9

1. Introducción

Localizer se utiliza desde Japón a Canadá, pasando por China, Rusia, Kazakstan, Ucrania, Francia, España, Alemania, Bélgica, etc..

Localizer es de hecho la herramienta de referencia para la construcción de aplicaciones web multilingües en la comunidad Zope. Proporciona servicios que van desde la creación páginas web en varios idiomas hasta la gestión de contenido multilingüe o la negociación de idioma.

1.1. Orígenes, el proyecto LLEU

El desarrollo del producto Localizer comenzó a principios del año 2001 en la Universidad Jaume I de Castellón (España). Se realizó en el marco del proyecto LLEU, con el objetivo de proporcionar la infraestructura necesaria para construir el sitio web de dicho proyecto en los dos idiomas oficiales de la universidad, el Catalán y el Castellano.

El proyecto LLEU se presentó en el III Congreso Hispalinux.

1.2. Esto es software libre

Localizer es cien por cien software libre, se distribuye bajo los términos y condiciones de la licencia GNU General Public License.

Su desarrollo sigue activo después de más de un año y medio, en torno a él se ha formado una pequeña comunidad de usuarios que contribuyen con ideas, documentación, parches, etc.. En el momento de escribir este artículo habían 45 personas suscritas a la lista de correo de Localizer y se habían realizado más de 800 "downloads" del producto.

1.3. Un producto Zope

Localizer está basado en el servidor de aplicaciones web Zope, el cual también es software libre. Tanto Localizer como Zope están escritos, principalmente, en el lenguaje de programación Python.

Zope es un entorno flexible que puede ser fácilmente extendido mediante módulos. Dichos módulos reciben el nombre de productos, Localizer es uno de esos productos.

Existen numerosos productos Zope que tratan de resolver el problema de la internacionalización y localización, de entre todos ellos Localizer se ha convertido en el más popular, prácticamente sin competencia. Como ejemplo, Localizer fue utilizado para la construcción del sitio web de EuroPython, el cual se creó con motivo del primer congreso europeo en torno al lenguaje de programación Python.

1.4. Contenido del artículo

El presente artículo se divide en dos partes, en las primeras secciones se mostrarán las diferentes funcionalidades que proporciona Localizer: los servicios para el desarrollo de productos Zope y de sitios web multilingües, la negociación de idioma, el soporte de Unicode y su habilidad o capacidad de ser fácilmente extendido.

A continuación se mostrarán algunos de los sitios web que han sido realizados con Localizer. Para terminar con las conclusiones del artículo.

2. Internacionalización y localización

Antes de empezar con el producto conviene explicar rápidamente dos conceptos fundamentales.

Internacionalización es el proceso mediante el cual una aplicación monolingüe, sin capacidad para mostrarse en varios idiomas, se transforma en una aplicación capaz de mostrar sus mensajes en diferentes idiomas, mostrar las fechas en los diferentes formatos locales, etc.. Es decir, se trata de todas las modificaciones que precisa un software para poder ser multilingüe.

Localización es el proceso por el cual se adapta una aplicación a los requerimientos locales de una determinada región. Por ejemplo, se proporcionan las traducciones de los mensajes, se especifica en qué formato se deben mostrar las fechas, etc..

La internacionalización es un paso previo a la localización que se realiza una sola vez, mientras que la localización se realiza una vez por idioma.

Como anécdota, comentar que existen dos abreviaturas para estos términos, la palabra internacionalización suele escribirse como "i18n" y la palabra localización como "l10n". Ambas abreviaturas se construyen del mismo modo (a partir del término inglés); la primera letra de la abreviatura es la primera letra del término, de forma similar para la última letra, el número que hay entre ambas es el número de caracteres que hay entre la primera y última letra del término.

3. Productos

Localizer no solamente proporciona facilidades para la construcción de sitios web multilingües (facilidades que veremos más adelante), también ayuda a desarrollar productos Zope en varios idiomas.

Un ejemplo es el propio Localizer, todas sus interfaces están internacionalizadas y localizadas en varios idiomas (español, japonés, alemán, vasco, húngaro, etc..).

La solución proporcionada por Localizer para esta tarea está construida en torno a las utilidades Gettext de GNU, las cuales son el estándar de hecho en la comunidad del software libre. Proyectos de considerables dimensiones, como GNOME y KDE, las utilizan.

3.1. Gettext

En Gettext se utilizan unas marcas especiales en el código fuente que identifican los mensajes que deben ser traducidos. Añadir estas marcas representa el paso conocido como internacionalización. La localización se realiza en varios pasos:

- El código fuente se analiza utilizando un programa de línea de comandos, el cual genera un fichero de texto con los mensajes a traducir, estos ficheros reciben el nombre de ficheros PO.
- Los ficheros PO generados (uno por idioma) son editados por los traductores humanos, quienes añaden las traducciones. Para esta tarea se puede utilizar cualquiera de los editores que existen (Emacs, KBabel, etc..).
- Finalmente, los ficheros PO se compilan para generar los ficheros binarios que se utilizarán en tiempo de ejecución. Estos ficheros binarios reciben el nombre de ficheros MO.

Las marcas utilizadas en el código fuente son en realidad llamadas a una función que recibe un mensaje y busca y devuelve su traducción a partir de los ficheros MO.

El uso de Gettext permite a Localizer aprovechar la experiencia y las herramientas desarrolladas en torno a él, esto incluye sofisticados editores de ficheros PO (para los traductores), soporte de formas plurales, etc..

4. Sitios web

Zope proporciona una interfaz web que puede ser usada para construir aplicaciones a través de la web. Esta interfaz es conocida como las pantallas de gestión, las cuales permiten gestionar los diferentes tipos de objetos que pueden usarse en Zope (plantillas, carpetas, imágenes, ficheros, etc..).

La mayoría de los productos Zope todo lo que hacen es proporcionar nuevos tipos de objetos a los ya disponibles en la distribución estándar. Estos objetos incluyen sus propias pantallas de gestión, las cuales se integran en el resto del sistema.

Localizer añade cuatro nuevos tipos de objetos, los cuales incluyen sus propias pantallas de gestión. Estos objetos proporcionan mecanismos para internacionalizar un sitio web, para gestionar contenido multilingüe o para personalizar la política de negociación de idioma. De este modo Localizer permite la construcción de sitios web multilingües a través de la web. Ahora veremos con mayor detalle estos cuatro tipos de objetos.

4.1. El catálogo de mensajes

El primer objeto que vamos a ver es el catálogo de mensajes, el cual, cómo su nombre sugiere, sigue una aproximación similar a la de Gettext. El catálogo de mensajes es una base de datos que almacena mensajes y sus traducciones, con una interfaz web para gestionarlas. Para obtener las traducciones proporciona una interfaz similar al de Gettext.

Para gestionar el catálogo, además de la interfaz web, es posible exportar e importar su contenido a y desde ficheros PO, de este modo se pueden utilizar sofisticados editores gráficos. Y todavía hay más, el catálogo de mensajes es accesible desde FTP, esto significa que los mensajes se pueden editar directamente utilizando un editor que soporte FTP, por ejemplo Emacs.

Los catálogos de mensajes se utilizan normalmente para internacionalizar la interfaz de usuario, por ejemplo el sistema de navegación, los formularios, etc..

4.2. Contenido

El contenido tiene unas necesidades diferentes y requiere por la tanto una solución diferente. Por ejemplo, el contenido debe ser indexado, de tal modo que se puedan realizar búsquedas sobre él; también debe ser posible utilizar herramientas de "workflow" para gestionarlo. Para satisfacer estos y otros requerimientos Localizer proporciona el objeto "Local Content".

La diferencia clave entre la solución para las interfaces y la solución para el contenido es que en el primer caso las traducciones se almacenan en una base de datos separada (el catálogo de mensajes) a la cual se accede cada vez que se precisa una traducción; sin embargo, en los objetos de contenido local las traducciones se almacenan en el mismo objeto que el texto original.

4.3. Imágenes, fechas, etc..

Todo lo que hemos visto hasta ahora se refiere al texto, bien para las interfaces o para el contenido. Pero hay más que texto en un sitio web, por ejemplo, ¿qué sucede si se precisa localizar una imagen o mostrar una fecha en el formato local?. Para estos propósitos Localizer proporciona el objeto "Local Folder", el cual es una solución genérica para localizar cualquier cosa.

Básicamente es una carpeta o directorio que puede almacenar diferentes versiones de cualquier tipo de objeto Zope y proporcionar la versión correcta (basándose en las reglas de negociación de idioma) cuando se requiere. Esta solución puede utilizarse para localizar imágenes, fechas, etc..

5. Negociación de idioma

En un sitio web multilingüe, cada vez que el cliente solicita una página web el servidor debe decidir en qué idioma servirla, esto se conoce como la negociación de idioma.

Por defecto un sitio web basado en Localizer tan solo considerará la configuración del navegador (la cabecera Accept-Language del protocolo HTTP) para elegir el idioma.

Pero el producto Localizer proporciona una clase (llamada también Localizer) que permite personalizar la política de negociación de idioma.

Esta clase proporciona varios métodos que ayudan a realizar las tareas más comunes, por ejemplo hay uno que almacena el idioma preferido por el usuario en una "cookie", y hay otro que muestra una caja de selección que permite cambiar dicha "cookie". También hay otro método que permite especificar el idioma en la url, la mejor solución para evitar problemas de cache y para los motores de búsqueda.

El desarrollador también puede escribir sus propios métodos para implementar la política que mejor satisfaga sus requerimientos.

Esta configuración o implementación de la política de negociación de idioma se realiza en el objeto Localizer, normalmente solo hay uno en cada sitio web. De este modo, es posible cambiarla en cualquier momento sin afectar al resto del sitio web, es decir, la negociación de idioma es ortogonal al resto del sistema.

Esto también significa que los desarrolladores de productos Zope ni precisan ni deben preocuparse por la política de negociación de idioma, ya que es algo específico a cada sitio web.

6. Unicode

El soporte de Unicode es una pieza central de Localizer.

El lenguaje de programación Python proporciona un tipo de cadenas de texto llamadas Unicode, las cuales ocultan los detalles de implementación relativos al sistema de codificación (en inglés "encoding"). Localizer utiliza este tipo de cadenas para almacenar el texto, esto simplifica considerablemente el trabajo del desarrollador.

Cuando las cadenas de texto Unicode se serializan, por ejemplo cuando el servidor envía una página web al cliente, se debe elegir un sistema de codificación. Qué sistema en concreto utilizar es una elección del desarrollador. Si la aplicación requiere combinar idiomas de diferentes alfabetos el sistema de codificación a utilizar debería ser UTF-8 o algún otro de la familia Unicode (UTF-16, etc..).

Como ejemplo, las pantallas de gestión de Localizer utilizan UTF-8. De hecho están traducidas al japonés y no hay ningún inconveniente en mezclar textos de diferentes idiomas.

7. Extensibilidad

Hemos visto rápidamente el conjunto de funcionalidades de Localizer que no requieren conocimientos de programación y que pueden usarse para construir sitios web multilingües sin una gran complejidad.

Pero eso no es todo, los desarrolladores pueden crear sus propios productos Zope que extienden Localizer utilizando la interfaz de programación que proporciona para estos propósitos, por ejemplo para construir objetos multilingües con características especiales.

De hecho, existen ya varios productos. Por ejemplo, los productos CMFLocalizer y PersistentTranslationService añaden varias funcionalidades relativas a otras tecnologías Zope (CMF y ZPT). Por otro lado, el producto Base18 explora nuevas estrategias para gestionar contenido multilingüe.

8. Ejemplos

En esta sección veremos brevemente algunos sitios web que han sido construidos con Localizer, y qué es lo que tienen de particular.

8.1. LLEU

El proyecto LLEU tenía por objetivo construir un sistema de información para los estudiantes de la Universidad Jaime I de Castellón. Dicho sistema debía estar basado en la web, es decir, debía concretarse en un sitio web.

La Universidad de Castellón se sitúa en una región bilingüe, tiene dos idiomas oficiales, el Catalán y el Castellano. En consecuencia el sitio web debía estar en ambos idiomas. Para conseguirlo, dada la ausencia de las herramientas necesarias, se desarrolló el producto Localizer, el cual proporcionó la infraestructura requerida.

Como características particulares de este sitio web destacaría:

- *Flujos de trabajo* (en inglés "workflows") de los documentos sensibles a los diferentes idiomas, de tal modo que cada versión puede encontrarse en un estado diferente. Por ejemplo la versión en Catalán podría estar revisada y lista para ser pública, pero la versión en Castellano podría estar todavía vacía.

Junto con este sistema se incluían interfaces de estadísticas que permitían desde un solo punto conocer el estado general del sitio web y de su contenido.

- Traducción automática. Gracias a la universidad de Alicante (proyecto Internostrum), se pudo ofrecer un sistema de traducción automática entre el Castellano y el Catalán, esto permitió reducir considerablemente los costes ya que ambos idiomas son relativamente parecidos y la herramienta proporciona traducciones bastante buenas que tan solo han de ser revisadas.

8.2. EuroPython.org

Este sitio web fue creado como el portal de la primera conferencia europea sobre el lenguaje de programación Python, celebrada en Charleroi en Junio del año 2002.

Lo que tiene de particular es que inicialmente se encontraba en un solo idioma, inglés, que fue posteriormente convertido en multilingüe utilizando Localizer. Es decir, no se construyó como una aplicación multilingüe desde el principio, si no que fue un paso posterior.

Otro aspecto significativo era el hecho de que el sitio web fue desarrollado por múltiples personas de tal modo que diferentes secciones del sitio utilizaban diferentes herramientas y tenían un estilo diferente. Además, el sitio web estaba ya en producción cuando el proceso de internacionalización comenzó. Todo esto suponía un reto para Localizer, un reto que superó con éxito.

9. Conclusiones

Como conclusión comentar que el producto Localizer sigue bajo activo desarrollo. Existen varias líneas de desarrollo.

En primer lugar, nuevas versiones de Zope y de otras herramientas y tecnologías relacionadas ofrecen nuevas posibilidades que Localizer debe explotar. En consecuencia existen varios proyectos para añadir nuevas funcionalidades, por ejemplo soporte de la sintaxis específica de internacionalización del lenguaje ZPT. En ocasiones este trabajo consiste en integrar funcionalidades ya presentes en los diferentes productos basados en Localizer.

Sin embargo, la línea de desarrollo principal consiste en posicionar a Localizer como una solución competitiva con respecto a las soluciones propietarias. Por ejemplo algunas de las funcionalidades a implementar son:

- Servicios de flujos de trabajo y de gestión versiones genéricos, no específicos para la gestión de contenido multilingüe, pero capaces de responder a las necesidades complejas de estos sistemas.
- Extracción de frases dentro de textos largos, alimentación de una base de datos externa con dichas frases, utilización de "fuzzy matching" o de otro mecanismo más complejo para obtener traducciones automáticas, desarrollo de interfaces sofisticados para los traductores. En definitiva, proporcionar las funcionalidades típicas de los sistemas de memorias de traducción.
- Soporte de formatos de ficheros, como HTML, PDF, etc.. De este modo los servicios de gestión de contenidos podrían utilizarse no solamente en ficheros de texto simples, si no también en textos con formato.
- Implementación de estándares como TMX, XLIFF, etc..

Finalmente, la última línea de desarrollo consiste en separar las partes de Localizer que no dependen de Zope en un paquete Python, exponiendo de este modo los servicios que Localizer proporciona a una comunidad más grande. Del mismo modo este producto podría inspirar o influir directamente a otros esfuerzos similares dentro de lenguajes o comunidades diferentes a la del lenguaje de programación Python y el servidor de aplicaciones web Zope.

Pero sin duda lo más importante es consolidar la comunidad en torno a Localizer, involucrando a más personas en el desarrollo del producto, sentando de este modo unas sólidas bases para llevar adelante los desarrollos arriba mencionados.

10. Referencias

1. El sitio web de Localizer, <http://www.localizer.org>
2. La Universitat Jaume I de Castellón, <http://www.uji.es>
3. El proyecto LLEU, <http://uji2001.lleu.org>
4. La licencia GNU GPL, <http://www.gnu.org/licenses/licenses.html#GPL>
5. El servidor de aplicaciones web Zope, <http://www.zope.org>
6. El lenguaje de programación Python, <http://www.python.org>
7. El portal EuroPython, <http://www.europython.org>
8. Las herramientas GNU Gettext, <http://www.gnu.org/software/gettext/>
9. El producto CMFLocalizer, <http://zope.org/Members/fafhrd/CMFLocalizer>
10. El producto PersistentTranslationService, <http://www.zope.org/Members/efge/PersistentTranslationService>
11. El producto Base18, <http://www.nexedi.org/software/Base18.stx>
12. El sistema de traducción Internostrum, <http://www.internostrum.com/>
13. El estándar TMX, <http://www.lisa.org/tmx/>
14. El estándar XLIFF, <http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm>