Interoperability Frankfurt-Madrid: ITS 2.0 CMS/TMS use case

Pedro L. Diez Orzas, Karl Fritsche, Mauricio del Olmo, Stephan Walter

[1] Linguaserve I.S. S.A., Seminario de Nobles, 4
28015 Madrid, Spain

60329 Frankfurt am Main, Germany
pedro.diez@linguaserve.com, Karl.Fritsche@cocomore.com,
mauricio.delolmo@linguaserve.com, stephan.walter@cocomore.com

Abstract
This paper is the description of how ITS 2.0 allows a better integration between a CMS and a TMS when translating content stored in a CMS, and how the localization workflow of the contents benefits from each ITS 2.0 data category implemented.

This use case has been developed during the MultilingualWeb-LT project that receives funding by the European Commission (project name LT-Web) and in the W3C MultilingualWeb-LT Working Group.

We will exemplify how the contents are generated in Drupal, a Content Management System (CMS). Before they are sent, the contents are annotated with ITS 2.0 metadata in two ways: automatic annotation and manual annotation. XHTML + ITS 2.0 is used as interchange format. Once created, they are sent to the Linguaserve Global Business Connector Server (GBC Server) translation server, processed in the Linguaserve internal localization workflow Platform for Localization, Interoperability and Normalization of Translation (PLINT). Afterwards, once the annotated content is translated and the metadata is treated, they are downloaded by the client and imported into the CMS. The ITS 2.0 selected data categories for integration are: Translate, Localization Note, Domain, Language Information, Allowed Characters, Storage Size, Provenance, and Readiness (ITS 2.0 extension).

Keywords: Multilingualweb, Internationalization Tag Set 2.0, ITS 2.0, web localization, interoperability, Content Management System, Translation Management System, metadata, web translation.

1. Introduction

The large volume of information and web content justifies the use of CMS systems for medium to large companies and organizations. They provide benefits as content control, several user profiles, abstraction and workflows.

When we introduce the multilingual variable to the CMS picture, a translation workflow is highly recommended. The advantages of using an external localization provider and computer assisted and automated Translation tools gives added value as the use of translation memories, glossaries and the experience with translation management.

This paper will exemplify how ITS 2.0 allows a better integration between CMS and TMS and how the localization workflow of the contents benefits from each implemented data category.

2 CMS and TMS Integration with ITS 2.0

In the setup described in this paper, Cocomore and Linguaserve have worked together with a real customer, the “VDMA - Verband Deutscher Maschinen- und Anlagenbau - German Engineering Federation” (www.machines-for-plastics.com/kug/). The languages combinations were German into French and into Chinese, and around 75,000 words were enriched with metadata, translated and processed.

The basic steps of an ITS 2.0-aware content creation and translation- process are as follows:

- VDMA has content produced in the Drupal CMS.
- Before being sent, the content is annotated with ITS 2.0 metadata by using automatic and manual annotation. This localization workflow is an XML based tool chain; hence, XHTML + ITS 2.0 is used as the interchange format.
- The content is sent to the Linguaserve Global Business Connector Server (GBC Server),
Localisation Focus

The International Journal of Localisation

Vol.12 Issue 1

processed in the Linguaserve internal localization workflow “Platform for Localization, Interoperability and Normalization of Translation” (PLINT).

• The ITS 2.0 metadata is used during the LSP (Linguaserve) internal processing for several localization tasks (providing context to the translators, blocking the non-translatable contents in the CAT tool, selecting terminology and translation memories…) and also updated in some cases as a result of the process (Provenance: the translator and proofreader that have done the job).

• Afterwards, once the annotated content is translated and the metadata is treated, they are downloaded by the client and imported into the CMS.

This integrated approach affects practically all areas of the traditional translation workflow. Accordingly, it requires modifications and extensions throughout the tool chain. Fig. 1 shows a vision of the architectural entities that are involved in ITS 2.0-aware content and translation handling.

3. ITS 2.0 Roundtrip

Some of the features of the solution that we created based on this architecture can be assigned to either the content provider’s or the LSP side of the picture.

3.1 Content provider’s side

On the content provider’s side the creation of the ITS 2.0 metadata aware workflow involves the following areas:

1. **Annotation of source language content with ITS 2.0 metadata within the Drupal CMS.** Structural annotation rules can be specified as global rules on a page/content type level, while local metadata is added by hand. In addition, automated annotation tools can be integrated through a standardized interface to support the user in creating such local markup.

Manual annotation features are available in all generally expected interaction modes (toolbar buttons, context menu, keyboard shortcuts).

Two annotation approaches are supported:

a) Annotation may be done as part of the content creation process, via features that have been added as plugins to the out-of-the-box Drupal WYSIWYG editor (see Fig. 2).

b) Annotation may be carried out as a separate step, without the ability to modify the content. This allows workflows that separate content know-how and translation management.

2. **Transparent data round-tripping**

Triggered from within Drupal, this is carried out in the background via export/import of files XHMTL+ITS 2.0 markup, to be automatically

**Figure 1. CMS-TMS ITS 2.0-aware architecture**

**Figure 2. Edit menu for local ITS markup**
sent to/received from the LSP. The process is based on an extended version of the Drupal translation Management (TMGMT)-module.

3 Translation review
IT 2.0 markup is retained in this step so that annotated information can be taken into account for QA purposes.

3.2 LSP side
On the LSP side, the creation of the IT 2.0 metadata aware workflow encompasses three areas:

1 Pre-production/post-production engine for processing content files annotated with IT 2.0.
2 LSP internal localization workflow to provide support to project management and production processes.
3 Computer Assisted Translation (CAT) tool usage for translation, proofreading and post-editing with IT 2.0 annotated content.

Fig. 3 illustrates the life cycle of each data category in the complete roundtrip.

4. IT 2.0 Implementation in the CMS
Cocomore integrated IT 2.0 into the open-source Content Management System (CMS) Drupal. This required the development and adaptation of several modules:

- Drupal TMGMT-module (extension to allow workflows with IT 2.0 annotation)
- Drupal WYSIWYG editor: Plugin for IT 2.0 annotation
- JQuery plugin for IT 2.0 annotation in a separate step (new implementation)
- Interfacing with Global Business Connector Contents (GBCC) and web services (implementation of data export/import and client implementation)

4.1 IT 2.0-aware translation workflow in the CMS

4.1.1 Workflow management with TMGMT
The workflow of translation and IT 2.0 handling within the open-source CMS Drupal can be done by extending Drupal with modules, and there are already a couple of modules available to help the user with translation processes.

We used and extended the “Translation Management Tool” (TMGMT). This module provides the basic translation workflow, which comprises the following steps:

Figure 3. ITS2.0 data category roundtrip
1. Create translation job
2. Send job to translation service
3. Reintegrate translated job into Drupal

Furthermore, TMGMT is designed to work with any content and any translation service. It provides interfaces for handling different sources and services. These are complemented by a default implementation of the source interface, which creates TMGMT jobs from content pages (called nodes in Drupal). For strings that are not part of Drupal nodes (such as menu links, error messages, etc.) we created an additional implementation, which generates a TMGMT job for the untranslated strings in the Drupal CMS, this module will be part of TMGMT in the future. To operate with the Linguaserve Web Service we implemented a translation service for TMGMT to send TMGMT jobs to Linguaserve and retrieve these jobs when they are finished. With these modules the general translation workflow is covered. The described functionality is implemented in the Drupal module TMGMT Workflow.

The TMGMT Linguaserve module which is a translation service for TMGMT handles all SOAP calls to Linguaserve and creates an XHTML file from a TMGMT job. This XHTML file is used as exchange format between Cocomore and Linguaserve. This file uses script-tags for global data categories and the normal HTML markup as described in the ITS 2.0 and only contains the content, no menu or styling information. In this way it can be easily interpreted by other services. The described functionality is implemented in module Drupal TMGMT Translator Linguaserve.

4.2 ITS 2.0 Annotation

4.2.1 Local markup via WYSIWYG
For the integration of ITS 2.0 we had to develop another module. This module provides the integration of ITS data categories into Drupal. It extends the WYSIWYG editor with new buttons to allow the user to add and edit local ITS markup in content pages. The following ITS data categories can be set with the WYSIWYG while creating or editing a content page:
- Translate
- Locale Filter
- Text Analysis
- Localization Note
- Language Information
- Directionality
- Terminology

The described functionality is implemented in module Drupal ITS 2.0 Integration.

4.2.2 Support for global markup
Apart from being able to set these data categories as local markup, there are also a few data categories that can act as global markup. Support for such global markup is managed on a per-content-type basis. Enabling ITS support for a given content type creates a new section in the edit form for content of this type. In this section, global XPath rules can be entered. It is possible to set default global rules for each content type or globally for the complete site.

For global markup the following data categories are available:
- Domain
- Translate
- Localization Note
- Revision/Translation Agent (from the Provenance data category)

The described functionality is also implemented in module Drupal ITS 2.0 Integration, but has to be enabled manually after installation.

4.3 Annotation as a separate workflow step

4.3.1 Functionality
In extension to the normal WYSIWYG editor in the content edit form we added a new “Language Management” form. The form provides an editor to only work on (add, remove change) the ITS 2.0 markup of a node, while the actual content is all write-protected. This supports a separation of content editing and ITS 2.0 annotation into two distinct workflow steps: A special user role (e.g. a translation manager) can add ITS data very easily after content creation without accidentally changing the content itself. This role will also be able to see and can edit the global markup.

4.3.2 User Interface
Local and global markup can be highlighted separately in the content. This is controlled in the UI by using checkboxes. In this way the user can choose what he wants to see and doesn’t get overwhelmed with all data categories at once. If the user selects content next to the selection, a small window pops up. In this window the user can choose a data category to add to the selected content. There are also keyboard shortcuts available for the data categories to support even faster tagging. For the simple data categories like “Translate” they just add the attribute with the most
contextually likely value. For instance, a translate attribute will be set to the negation of the value pertaining to the current context, thus translate=”no” within text that is not in the scope of any other translate attribute (because the default translate=”yes” is assumed for such text). For all other data categories a new modal window appears where the needed data can be edited, like the note and note type for “Localization Note”.

4.3.3 Implementation
The functionality described above depends highly on JavaScript and is built on top of the ITS 2.0 jQuery plugin, which was also developed by Cocomore. This jQuery plugin provides a functionality for the selection of text nodes with special data category values, and for getting the ITS values of a text node. It is released independently of the Drupal modules. Thus other frameworks or users can use it in their implementations as well. For example a programmer can quickly get all non-translatable text nodes of a HTML and XHTML page to add special styles to it. The plugin correctly handles both local and global markup, including global markup in a script tag and external linked global markup. The module performs all the ITS 2.0 tests.

4.3.4 Data categories with automatically determined values
There are several data categories that have a special status when integrating ITS 2.0 in a CMS due to the fact that they allow for an especially high degree of integration. This may be because the CMS provides specific means for handling them out of the box, or because adequate values for them can be derived automatically from other information that is available from various sources within the CMS and workflow. This special status is also reflected in the Drupal ITS 2.0 integration module.

For the Domain data category you can select that the area where the user can type in the domain shouldn’t be a text field, instead you can use the taxonomy system from Drupal. With this you can create your own vocabulary or use an existing one and just select the domains on content editing. The Provenance data can’t be edited by the user, it just shows and stores this information and it will be automatically set by the translation service. In a similar manner, additional data categories are embedded in the translation process. Data categories like Allowed Characters, Storage Size and Readiness from the ITS extension will be added automatically to the content sent to the LSP depending on Drupal’s field definitions of a particular field. As an example, there is a maximum length of 255 characters for the title field, and in this case the storage size category is added to the title field with the respective values set. The user doesn’t have to care about this at all. As another example, the expected finalization date and priority are added by the translation manager before the translation job is submitted to the LSP.

The described functionality is implemented in module Drupal ITS 2.0 Integration.

5. ITS 2.0 Implementation in the TMS
This section explains which ITS 2.0 data categories have been implemented, their usage and application on the different phases of the localization workflow.

There are also explanatory details on their implementation and examples of ITS metadata.

The Java classes involved in the ITS 2.0 processing of the contents are three:

• To manage the paths of the files and the data base records.
• To parse the documents and traverse the nodes in the pre-production and post-production phases.
• To provide the methods related with ITS 2.0 data categories integration.

A general view about the use of each data category in the Linguaserve localization workflow is shown in tables 1, 2 and figure 4.

Figure 4. Format Transformation Workflow
### Table 1. Data Category Treatment in the Internal Pre-Production Phase

<table>
<thead>
<tr>
<th>Data category</th>
<th>L10N workflow</th>
<th>XHTML Global</th>
<th>XHTML Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate</td>
<td>-</td>
<td>Omit selected not translatable contents.</td>
<td>A particular node could be not translatable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mark parts of the content marked as not translatable for blocking.</td>
</tr>
<tr>
<td>Localization Note</td>
<td>When alert type, send a notification to the project manager and add tooltip visualization in the workflow.</td>
<td>Create reference node to inform the translator.</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Domain</td>
<td>Automatic selection of terminology and translation memories.</td>
<td>Create reference node to inform the translator.</td>
<td>-</td>
</tr>
<tr>
<td>Language information</td>
<td>Quality check to ensure the source language content is according to the webservice parameter.</td>
<td>-</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Allowed Characters</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Storage size</td>
<td>Quality check for the original content.</td>
<td>-</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Provenance</td>
<td>Possibility to reassign the same translator/proofreader in new versions of the same content (based on identifiers).</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Readiness (*)</td>
<td>Priority checked with webservice.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 2. Data Category Treatment in the Internal Post-Production Phase

<table>
<thead>
<tr>
<th>Data category</th>
<th>L10N workflow</th>
<th>XHTML Global</th>
<th>XHTML Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate</td>
<td>-</td>
<td>Omit selected not translatable contents.</td>
<td>A particular node could be not translatable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mark parts of the content marked as not translatable for blocking.</td>
</tr>
<tr>
<td>Localization Note</td>
<td>When alert type, send a notification to the project manager and add tooltip visualization in the workflow.</td>
<td>Create reference node to inform the translator.</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Domain</td>
<td>Automatic selection of terminology and translation memories.</td>
<td>Create reference node to inform the translator.</td>
<td>-</td>
</tr>
<tr>
<td>Language information</td>
<td>Quality check to ensure the source language content is according to the webservice parameter.</td>
<td>-</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Allowed Characters</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Storage size</td>
<td>Quality check for the original content.</td>
<td>-</td>
<td>Inform the translator.</td>
</tr>
<tr>
<td>Provenance</td>
<td>Possibility to reassign the same translator/proofreader in new versions of the same content (based on identifiers).</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Readiness (*)</td>
<td>Date control for availability and delivery.</td>
<td>Update the data category node.</td>
<td>-</td>
</tr>
</tbody>
</table>
5.1 Use of ITS 2.0 data category

5.1.1 Translate

a) Translate in the pre-production phase
A method that obtains the ITS global rules and another method that obtains the global translatable rules from the ITS global rules were implemented.

After that, the global translate rules (translate="yes") and the global non-translate rules (translate="no") are stored in two different objects. The document nodes are traversed and for each node:

If a global translate rule applies to the node (xpath) then the current state of translate is updated for direct application and inheritance. In the local translate rules, the current state of translate and the defaults are checked to know the treatment of the node. The current state is also accordingly updated for inheritance: if the node is not translatable, jump to the next node; else, if the node is translatable, mark the node as translatable, then extract the content.

For local rules application, if any, traverse the HTML content, add tags for blocking content with translate="no" in the CAT tool, and put the content in the CAT tool oriented XML in a translate node. See example 1.

b) Translate in the post-production Phase
First, we traverse the nodes of the document, if the node was marked as translatable, besides recuperate the translation from the translated CAT tool oriented XML.

If there is HTML mark-up in the content, remove the marks for blocking non-translatable parts and insert the translation in the document. See example 2.

5.1.2 Localization Note

a) Localization Note in the pre-production phase
A method obtains the ITS global rules, while another method obtains the global localization note rules from the ITS global rules. A third method obtains all the localization notes of alert type.

If there is at least one alert type localization note, an e-mail is sent to the project managers and the comments of the file are updated in the database of the system for tooltip visualization in the localization workflow. After that, the document nodes are traversed and for each node:

If a global localization note rule applies to the node, then a reference node is created in the CAT tool oriented XML for the translators/proofreaders. See examples 3 and 4 (global and local usage).

5.1.3 Domain

a) Domain in the pre-production phase
A method obtains the ITS global rules, another method obtains the global domain rules from the ITS global rules, and a third one stores the domains associated with the file in the system’s
database.

In this way, the document nodes are traversed and for each node:

If a global domain rule applies to the node, then a reference node is created in the CAT tool oriented XML for the translators/proofreaders. See example 5.

b) Domain in the CAT tool project creation step

The domains associated with each selected file are retrieved and listed.

The dictionaries corresponding with each domain are obtained and associated to the CAT tool project. In this manner, the CAT tool can then use the dictionaries selected based on the domain values.

c) Domain in the CAT tool project export step

When the translation and proofreading tasks have ended in the CAT tool, the files are exported. In this step, when the CAT tool project is closed, the memory files are stored in the paths corresponding with each domain. The translated files advance in the localization workflow to the post-production phase.

5.1.4 Language Information

a) Language Information in the pre-production phase

The document nodes are traversed and for each node:

The paths of the translation memories corresponding with each domain are obtained.
If there is language information, it is checked to see if it is the same than the source language information declared in the system; if not, a warning for the project manager is shown in the workflow. See example 6.

b) Language Information in the post-production phase
The document nodes are traversed and for each node:

If the node has language information, update the value of the original language code with the target language code. The same process is made in the contents with HTML, but only within the parts that have been translated. See example 7.

5.1.5 Allowed Characters
a) Allowed Characters in the post-production phase
Here, the document nodes are traversed and for each node:

If the allowed characters restriction is declared, it is checked with the regular expression, but if the restriction is not fulfilled, an exception is raised, the process is aborted and the user is informed about the reason. See example 8.

5.1.6 Storage Size
a) Storage Size in the pre-production phase
We traverse the document nodes and for each node:

If the node is translatable and has storage size limitation declared, a method checks the maximum storage limitation compliance, for which it also takes into account the encoding declared for the content. See example 10.

b) Storage Size in the post-production phase
In the post-production phase, if the node is translatable and has storage size limitation declared, the maximum size is informed in an attribute of the translatable nodes of the CAT tool oriented xml. The size of the original content (in another attribute) is also reported. This information will be available for the translators/proofreaders in the CAT tool. The size is calculated using the encoding.

It is also checked if the original content fulfills the restriction and, if not, a warning is shown to the project manager. See example 9.

5.1.7 Provenance
a) Provenance in the pre-production phase
Here, for each node, if there is provenance information available from a previous translation, the database is updated to register the translator and the language pair. On the other hand, if there is provenance information available from a previous proofreading, the database is updated to register the proofreader and the language pair. See example 11.
b) Provenance in the translation CAT tool phase
The system proposes the project manager the last translator who performed the same task for the same language pair.

Example 10

Example 11

Traductor
French translator 21686

Figure 6. Page of CAT Workflow - Translation

The expected delivery date is updated in the system, taking into account the time zone, and the priority of the translation is checked with the information available in the system. If there is no concordance, a warning for the technical department is shown in the workflow. See example 12.

5.1.8 Readiness (ITS 2.0 Extension)

a) Readiness in the pre-production phase
A method obtains the ITS global rules and another method obtains the global readiness rules from the ITS global rules were created.

The expected delivery date is updated in the system, taking into account the time zone, and the priority of the translation is checked with the information available in the system. If there is no concordance, a warning for the technical department is shown in the workflow. See example 13.

Example 12

Example 13
6. Links to Information

6.1 Drupal documentation of components

All the implementations are released under the GNU General Public License 2 and can be downloaded and modified.

They are available at the following URLs:


The ITS-Drupal module uses the ITS 2.0 jQuery Plugin, which we published separately for users who do not use Drupal as their CMS, but want to work with ITS 2.0 in an HTML context too. This plugin is tested with the W3C ITS 2.0 Test-suite and conformant to the standard.

Our Drupal implementation is extensible with other modules. Cocomore developed an interface to allow other systems to do work before or after a translation. This can for instance be used to add a QA service after the translation is done or to integrate a service for additional automatic annotation of ITS 2.0 metadata. An implementation that integrates an Enrycher

Example 14

If the file is processed after the expected delivery date, a warning for the project manager is shown in the workflow. See example 14.


Figure 8. Pre-production/post-production engine for Drupal XHTML files with ITS 2.0

<table>
<thead>
<tr>
<th>File to upload:</th>
<th>Browse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Size</td>
</tr>
<tr>
<td>Example: 193_demo Proc. xml</td>
<td>243</td>
</tr>
<tr>
<td>Example: 193_demo proto xml</td>
<td>749</td>
</tr>
</tbody>
</table>

The ITS-Drupal module uses the ITS 2.0 jQuery Plugin, which we published separately for users who do not use Drupal as their CMS, but want to work with ITS 2.0 in an HTML context too. This plugin is tested with the W3C ITS 2.0 Test-suite and conformant to the standard.

Our Drupal implementation is extensible with other modules. Cocomore developed an interface to allow other systems to do work before or after a translation. This can for instance be used to add a QA service after the translation is done or to integrate a service for additional automatic annotation of ITS 2.0 metadata. An implementation that integrates an Enrycher

Example 14

If the file is processed after the expected delivery date, a warning for the project manager is shown in the workflow. See example 14.
service to generate text analysis markup is accessible at the following URL:


6.2 Videos and demo of the TMS Processing


Acknowledgements

Many thanks to the MultilingualWeb-LT project (funded by the European Commission, through the Seventh Framework Programme in the area of Language Technologies, project name LT-Web, number FP7-ICT-2011-7-287815), to the W3C MultilingualWeb-LT Working Group, and to the Internationalization Tag Set Interest Group. Also, thank you very much to Laura Guerrero, Giuseppe Deriard, Pablo Badía, and Felix Fernández (from Linguaserve), and Clemens Weins (from Cocomore), who participated in this use case.

References


