D3.1.5: REPORT ON LT-WEB PROCESSING IN THE CMS

Karl Fritsche, Stephan Walter

Distribution: Public

MultilingualWeb-LT (LT-Web)
Language Technology in the Web

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	3.1.5
Deliverable title:	Report on LT-Web Processing in the CMS
Dissemination level:	Public
Contractual date of delivery:	31st September 2013
Actual date of delivery:	9th December 2013
Author(s):	Karl Fritsche, Stephan Walter
Participants:	Cocomore
Internal Reviewer:	
Workpackage:	WP3
Task Responsible:	Clemens Weins
Workpackage Leader:	Clemens Weins

Revision History

Revision	Date	Author	Organization	Description
5	06/12/2013	Stephan Walter	Cocomore AG	Finalization
4	17/09/2013	Karl Fritsche, Stephan Walter	Cocomore AG	Complete re-write in best practice-style.
3	19/07/2013	Karl Fritsche, Stephan Walter	Cocomore AG	Added Introduction to every chapter.
2	02/07/2013	Yves Savourel	ENLASO	Added deliverable IDs and initial 3.1.4 content
1	02/07/2013	Karl Fritsche, Stephan Walter	Cocomore AG	First Structure Draft

CONTENTS

1. Executive Summary

This document discusses best practices to consider when working with ITS 2.0 inside a content management system (CMS). Working with ITS 2.0 inside a CMS can mean adapting or even implementing a CMS so that it supports ITS 2.0 metadata in content or via its translation facilities. It can also mean using such an adapted CMS in practice to handle multilingual content. Accordingly the document is divided in two sections dealing with these two main topics. The document is informed by the experiences gathered in the MultilingualWeb-LT project, where an ITS 2.0 aware translation workflow was implemented within the open source CMS Drupal.

This deliverable is a snapshot of the "ITS 2.0 in the CMS - Best Practices guide" provided to the community in the ITS IG Wiki at http://www.w3.org/International/its/wiki/ITS_2.0_in_the_CMS_-_Best_Practices as a living document. This will allow the best practices presented to be extended and adapted continuously with growing adoption of and experience with the standard.

2. Introduction

This document is a complement to the W3C Recommendation Internationalization Tag Set (ITS) Version 2.0. It discusses topics that occur when using ITS 2.0 in connection with a CMS, and suggests ways to deal with these topics. The document is informed by the experiences gathered in the MultilingualWeb-LT project, where an ITS 2.0 aware translation workflow was implemented within the open source CMS Drupal. Its aim however is to provide guidance independent of one particular CMS as far as possible. An important aspect is therefore how the different characteristics of CMSs interact with ITS usage and handling.

However, not all internationalization-related issues can be resolved by the special mark-up described in ITS 2.0. The best practices in this document therefore go beyond application of ITS mark-up to address a number of problems that can be avoided by correctly designing the XML format, and by applying a few additional guidelines when developing content.

This document and Internationalization Tag Set (ITS) Version 2.0 implement requirements formulated in the W3C Working Draft Requirements for Internationalization Tag Set (ITS) 2.0.

This set of best practices does not cover all topics about ITS 2.0 in the CMS. Other useful reference material includes the deliverables D2.1, D3.1.1, D3.2.2, and D5.1.1 of the MultilingualWeb-LT project.

2.1. Who should use this document?

This document is divided into two main sections:

1. The first section addresses developers and architects implementing based on a CMS, or even implementing a CMS. It deals with the technical questions of how to handle ITS 2.0 metadata in the CMS and how to design the interaction (workflow, interfaces) with other systems (translation tools on the side of the language service provider, analysis tools, etc.).
2. The second takes the perspective of CMS users, in particular the "content worker" roles of authors, reviewers and translators. It discusses topics like entering or accessing the information in the mark-up (UI level), and which information is needed by whom at which step in the translation process (workflow and process level). For this purpose it will also have to touch on some more general questions of localization/translation with a CMS.

2.2. How to use this document?

A few words of *caveat lector*:

- Many of the best practices discussed here deal with very specific questions that are much better understood against the background of a more general picture. For this reason both sections (Developers and CMS Users) contain a first entry that gives you such a picture. You are invited to consult these entries for some broader context on how things were approached in the MultilingualWeb-LT project.
- Many of the decisions making up each of the best practices provided here depend on other decisions that may or may not be looked at in the context of other entries. We tried to mention such dependencies (and provide links to make them explicit), but probably did not always do so. Also note that conflicting goals may lead to inconsistencies between some of the best practices. Keep in mind that they are meant as advice from experience, not as definitive answers.
- Since ITS 2.0 is still very young and the partners in MultilingualWeb-LT did pioneering work in its implementation, many of the best practices are actually based on common sense paired with one single observation. They are therefore likely to change (for the better) with broader adoption and growing implementation experience.
- While the two sections of this document address distinct groups of users (Developers and CMS Users) each group's section may be of interest for the other one as well. In particular developers are likely to be confronted with the considerations in the CMS Users' best practices sooner or later in the form of business requirements.

3. DEVELOPERS

3.1. A general picture of an ITS 2.0-enabled CMS and its interaction with a TMS

As a preface and background to the more specific best practice items in the following sections, the diagram in Figure 1 gives an overview of the general architecture adopted in the reference implementations created in the MultilingualWeb-LT-project (creating ITS 2.0-aware translation facilities inside Drupal and integrating with the Linguaserve PLINT translation environment):

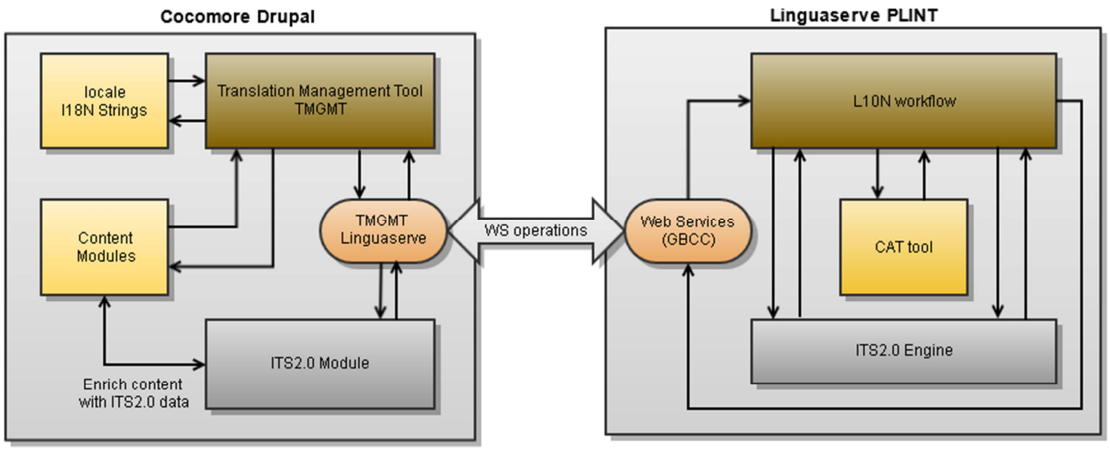

Figure 1: ITS 2.0 aware CMS-TMS interaction in MultilingualWeb-LT

Focussing on the CMS side of the picture:

- the solution built upon Drupal's module system and was incorporated into several Drupal modules
- it set up upon a community module for translation management (TMGMT, https://drupal.org/project/tmgmt)
- it provides annotation capabilities inside Drupal's content editing facilities (WYSIWYG editors), as well as an additional Javascript based annotation tool that protects the page content itself from modifications
- UI strings, menus etc. can also enter into the ITS 2.0 aware translation workflow
- LSP interaction is handled through an adaptor encapsulating a SOAP interface (again realized as a Drupal module)

3.2. Take the CMS internal data representation into account when integrating ITS

To use ITS 2.0 mark-up in your content (definitely for local mark-up, but basically also for global mark-up) it is a great advantage if the CMS in use is using a HTML or XML data representation, and this may be an important criterion for choosing a CMS if you have the need for an ITS 2.0 aware translation workflow as part of your use case. Otherwise you can't add ITS2.0 data categories locally, and also it is not possible to point global rules to non XML or HTML structures via XPath or CSS selector. It is possible to convert ITS2.0 data categories from HTML to XML and back in case the CMS saves the data as XML, see https://github.com/kosek/html5-its-tools. For other mark-up languages like a wiki syntax or markdown it is hard to use ITS2.0, as you sometimes need to add multiple attributes and value pairs, which is often not supported by such languages. Should your CMS use some such kind of representation (or even something completely different, like a decomposition of text content and markup into a relational data model or a completely proprietary binary format) it may be advisable to implement a mapping to HTML or XML as an intermediate stage, and implement ITS 2.0 handling on top of this.

3.3. Try to build up on top of the editing facilities of your CMS for ITS annotation

The best way is to use the HTML syntax for ITS 2.0 and integrate it directly in the content editor of your CMS as much as possible. If the HTML editor in your CMS is configurable or extensible it is a good idea to make ITS annotation functionalities available in all expected interaction modes (be it through a menu, ribbon, context menu or keyboard shortcuts), but separate them clearly from the native functionalities of the editor (e.g. as a separate group of buttons). Given the relation of most data categories to text spans, annotation should work with all modes of text selection that are supported by the respective editor.

As an example of how this may be approached take the reference implementation for Drupal's WYSIWYG-editor provided in MultilingualWeb-LT and shown in the screenshot in Figure 2.

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

- Annotation consists simply in selecting a text span and assigning it the right category/attributes by one of these modes
- Controls for annotation are visually distinct from standard controls of the Drupal WYSIWYG-editor.

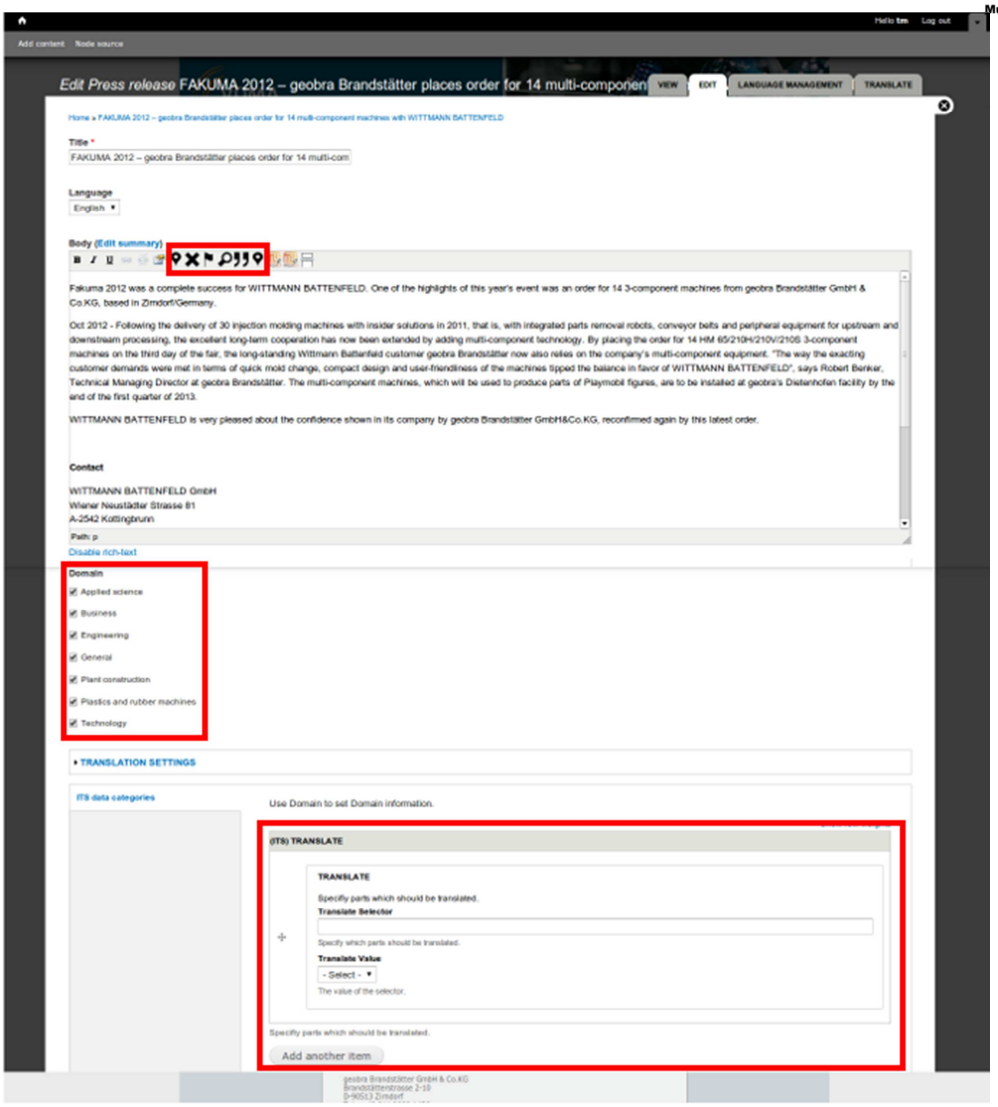

Figure 2: Editing ITS 2.0 metadata in a Drupal WYSIWYG editor

3.4. Build up on generic means for programming metadata editing facilities as far as possible when you do not use the editors that come with the CMS

To ensure a maximum of re-usability and browser independence it is advisable to work with generic means when implementing metadata editing outside the CMS editors. Ideally, a solution:

1. works inside the browser
2. makes no use of browser specific features or provides fallbacks if this cannot be avoided
3. encapsulates CMS specific functionality so that it can be re-implemented for other CMSs without affecting other parts of the solution

Thus a browser-based solution is preferable to a fat client on the local desktop. A solution that is based on a Javascript-framework (e.g. jQuery) has advantages over a plugin that is tied to a specific API. And using the adaptor pattern to hide away the CMS specific data storage and retrieval logic behind a generic interface make a solution re-usable across CMSs.

As an example the ITS 2.0 reference implementation stack created within MultilingualWeb-LT comprises a jQuery plugin, see http://plugins.jquery.com/its-parser/, that parses and allows handling/editing ITS 2.0 mark-up inside the browser by interacting with the DOM of a document created by Drupal, and storing the changes back to the Drupal database.

3.5. Leave ITS 2.0 metadata in the HTML that will be sent to clients if possible

The mark-up for the ITS 2.0 data categories has no impact on the design of a page, so it is not necessary to strip the corresponding attributes out. Moreover it may in fact make sense to leave annotated metadata in the HTML that your CMS delivers to clients even if it is only intended for the internal translation process. This information can be helpful in unanticipated use cases. It will for instance (at some point in the adoption of the standard) help automatic translation software like the built-in tools in Chrome to make the translation better, or help screen readers to change pronunciation for marked words. Our recommendation is therefore to deliver ITS 2.0 metadata to the users as the default case and only filter if there are any compelling reasons (like non-disclosure of internal information).

If you want to output the ITS 2.0 metadata to the web and not only use it for the internal translation process, you need to allow the HTML attributes for your HTML filters so they don't get stripped out. If your CMS supports the configuration of HTML filters with a regular expression like syntax, you could use an expression like the following to allow all local ITS metadata in HTML:

```
*[its-allowed-characters|its-annotators-ref|dir|domains|its-within-text|lang|its-locale-filter-list|its-locale-filter-type|its-loc-note|its-loc-note-ref|its-loc-note-type|its-loc-quality-issue-comment|its-loc-quality-issue-enabled|its-loc-quality-issue-profile-ref|its-loc-quality-issue-severity|its-loc-quality-issue-type|its-loc-quality-rating-score|its-loc-quality-rating-score-threshold|its-loc-quality-rating-vote|its-loc-quality-rating-vote-threshold|its-loc-quality-rating-profile-ref|its-mt-confidence|provenanceRecordsRefPointer|its-storage-size|its-storage-encoding|its-line-break-type|its-term-confidence|its-term-info-ref|its-term|its-ta-class-ref|its-ta-confidence|its-ta-ident|its-ta-ident-ref|its-ta-source|translate]
```

3.6. Provide support for global rules and apply them automatically to reduce manual effort if categories can be annotated based on structural regularities alone

Specific content may always have the same structure and certain ITS 2.0 data categories (e.g. translate for translatable/non-translatable content) may therefore be assignable by purely structural rules. ITS 2.0 allows you to use global rules in this situation, so that you do not have to repeat the same mark up for every piece of content. Global rules can be provided with the context or linked in from other sources. ITS 2.0 defines a default mechanism to resolve conflicts in the interaction of global and local mark-up.

Correspondingly, an ITS 2.0 enabled CMS should ideally support linking to global rule files as well as the dynamic creation of global rule files or other means of defining global rules within the CMS.

3.7. Provide support for configuring global rules at all necessary granularities

Depending on the kinds of use case and data it may make sense to provide global rules for content on varying levels. It may be a good choice to specify rules for all content, for certain kinds of content or for individual content items. It is therefore advisable to allow for configuration and overriding of global rules on all levels of

granularity or abstraction that the individual CMS provides.

As an example the Drupal CMS ITS 2.0 Integration module created as a reference implementation within MultilingualWeb-LT allows you to configure global rules on three levels:

1. create site wide defaults
2. specify per content type global rule files
3. change global rules for every page individually if needed

3.8. Try to make best use of out-of-the-box CMS features on data level

In many different CMS there are features that directly provide information for certain ITS 2.0 data categories, that can be used to determine such information with little effort, or that allow you to enter and store such information 'natively' within the CMS. It is advisable to capitalise on this fact as much as possible when integrating ITS 2.0 capabilities in the CMS:

- to reduce the manual annotation workload (automatic determination of values)
- to avoid unnecessary implementation effort (re-use of existing means to enter data)
- to increase usability and consistency by using only one way for entering one piece of information (re-use of existing means to enter data)

Some examples of data categories that benefitted from this approach in the context of the Drupal-ITS2.0 integration created as a reference implementation within MultilingualWeb-LT:

- **Domain:** The Domain data category is used to categorize your data and to set the domain of the text. Often CMSs provide means for "tagging" content. This mechanism can be used to identify your domains and set the correct data for this category.
- **Allowed Characters:** In some CMSs it is possible to define the type of a single input field like title, description, price, date or similar. This information can be used to generate information on allowed characters. Dates for instance may contain only numbers and some special characters allowed, and you can use "[0-9.-/]" for this. Depending on the system and available type you can define this once and use it automatically for all of your content.
- **Storage Size:** If the CMS uses a database to save the content you will almost inevitably have restrictions on how long special texts like titles can be. From the schema of these fields you can get the storage size and set it for all of your content.
- **Language Information:** Many CMS have the ability to handle text in different languages, and since the Language Information data category uses the already existing attribute *language* in HTML, in most of the cases you don't have to implement this manually. If there are multiple languages on one page you should add this data category separately and can retrieve the information from the content. The same applies for the data category Directionality.

3.9. Interface with automatic annotation tools to reduce manual annotation effort

Manual annotation (specifically for local annotation of mostly content-dependent data categories such as *Text Analysis* or *Annotation*) may be greatly speeded up by (or sometimes even require) integration with automatic annotation tools. When enabling a CMS for use with ITS 2.0 this topic should be given thorough consideration from the beginning of the project. Clearly most decisions that need to be taken here depend almost inevitably on the 'individual circumstances' of the respective CMS and tools in question. However there are certain

general questions that will typically have to be answered in one way or another:

- **Possible integration points:** Where within the workflow/data flow/logic of the CMS is it technically possible to integrate with a given tool?
- **Reasonable integration points:** At what stage or point in the content lifecycle does it make sense to integrate? (When is the information needed most urgently? When are the pre-conditions for using the tool fulfilled?) For instance it may not make sense to call a terminology identification tool before the domain of a text has been entered in the CMS.
- **Integration mode, protocols etc.:** It will always be preferable to interface via well-defined standards and interfaces (SOAP, REST) or at least use documented APIs, as well as open and accepted standards (XLIFF, XHTML) for encoding the data being transported, if only to ensure maintainability and to be future-proof. Ideally the tool one is interfacing with will even support ITS 2.0 itself. In reality of course one cannot choose what the tool at hand supports and does. This means that one may be forced to accept a home-grown data format based on the tools capabilities and implement an adaptor to create ITS mark-up from this. In any case considerations on data formats and protocols should not be left out of the picture when choosing a tool to integrate with. They may sometimes even outweigh linguistic result quality to a certain degree.
- **Task specific pre-conditions of the tool:** Which information does the tool need in order to deliver optimal results (e.g. a certain context window, domain information, training data, user feedback, ...)
- **Usability for the CMS user:** How should the tool's result be presented to the CMS user? Especially if a tool is expected to deliver imperfect results that require user actions (review, choosing from alternatives, etc.) it may be an option to present them interactively in the annotation UI.

3.10. Decide on essential features of your translation workflow when planning ITS 2.0 integration

There are a few absolutely essential decisions regarding the translation workflow that will shape how to approach ITS 2.0 integration into a CMS from the very beginning:

1. Will translation happen inside the CMS or inside another system (such as the translation management system of a language service provider)? In the first case, there will be a lot of UI level implementation work to be done regarding presentation of and interaction with metadata during the translation. In the second case, there will be important issues regarding ITS-handling at the interface to the other system.
2. Should there be support for multiple competing language service providers, freelance translators, etc., or is the set-up a cooperation with one specific vendor? In the first case there will be a focus on providing ITS 2.0 metadata with content for cost-estimates, selecting providers based on their ITS 2.0 capability, etc. that will not be needed in the second case.
3. Are there any restrictions on who may do what during the content creation and annotation process? For instance there may be a need to separate annotation from content editing, requiring an annotation UI tool that is separated from the content editor of the CMS. Or it may be desirable to have both tasks done by the same people using the same tool, requiring annotation features to be integrated into the content editor.

The scenario handled by the reference implementations created within MultilingualWeb-LT and investigated in the real life tests in the project was one where content could be annotated by two different user roles: Once during creation in a Drupal WYSIWYG editor and again afterwards in an editor that supported only modification of metadata but not of content. Content was then sent to an ITS 2.0 aware language service provider, who was essentially assumed to be the only vendor to be made available in the workflow. The

underlying infrastructure is based on the Drupal TMGMT module and is able to handle an arbitrary number of translation providers, with additional implementation requirements for the respective connectors.

3.11. Try to work with and extend existing internationalization and translation management features of the CMS

Many CMSs provide some level of translation and internationalization support. This may for instance cover all or some of the following:

1. Handling source content and its translations as a unit (or as related items), together with elementary information such as language, character set, etc.
2. Handling basic meta-information for translated content, such as version of the original, date of the translation, translator
3. Handling status and version information (item needs re-translation because source has changed, how many target languages are covered, how many translations have been published, interaction between versioning of source and translations)
4. Workflow features (send to translation, due dates, review features, ...)

The degree of support differs a lot across CMSs, and there are of course many different approaches, and even more differences on the technical level. When planning the integration of IT 2.0 support into a specific CMS it is therefore important to analyse the situation for that CMS with respect to the aspects listed above. One reason is obviously that a lot of work can be saved by re-using as much as possible of what has already been done by others for that specific CMS. Two other reasons relate more to questions of user acceptance and usability: Firstly if there is a translation solution in use by the mainstream of CMS users, acceptance will be a lot higher if the newly integrated features fit into this solution than if they work against it. Secondly, as discussed above (3.8), some of the internationalization features of a CMS may help to save annotation effort by providing values for some of the ITS 2.0 data categories directly.

Drupal provides basic internationalization support (covering the topics mentioned in the first point of the list above) out of the box. A community module for translation management (TMGMT) allows to define translation workflows (including review steps and status overview features) and to control the interaction with language service providers as well as automatic translation services. Within the MultilingualWeb-LT project we built upon this module for the ITS 2.0 enabled translation features for Drupal that were created as a reference implementation.

3.12. Try to integrate with workflow features and tools of the CMS

It is good practice to map workflow-related requirements to native features of the CMS as far as possible. Regarding topics related to the workflow itself (e.g. check out/check in-mechanisms, edit-review/approval-/publish-sequence) will guarantee that the ITS 2.0 aware translation workflow integrates seamlessly with the known concepts in use in the CMS. Besides such acceptance and usability benefits this approach will also reduce the risk of inconsistencies with non-translation related workflow features on the technical level. For instance by using the same CMS-provided concepts for edit-review-approval-mechanisms in general content creation and translation, it may be easier to deal with dependencies between the approval status of a translation and its source content than if both items are subject to different mechanisms.

Also in the case of roles it is useful to build on CMS-provided possibilities whenever the option exists. For instance assume that it is possible to define roles with specific rights per workflow step, you require a role that may only annotate ITS 2.0 metadata but not modify content and you have editing tools that allow annotation without touching content. Then you may want to define a workflow step for annotation based on your

annotation tool, create an "annotator" role and give it permissions only to see documents in this specific step.

Such considerations may become more or less obsolete if it is possible to set up on native internationalization and translation features of the CMS. For instance in the case of the Drupal reference implementation in MultilingualWeb-LT the Drupal TMGMT module provides basic workflow functionality, while Drupal provides role management. These features could be combined with a custom annotation tool to create a dedicated annotator-role with appropriate permissions and capabilities.

A different area where it may be beneficial to re-use what is already provided by a CMS consists in tools that become relevant in the context of workflow management. Examples are a tree view for visualizing a content item's editing history, a diff viewer for pointing out changes made in a workflow step, or a dashboard for showing a summary of status in ongoing translation workflows.

4. CMS USERS

4.1. A possible general picture of translation with a CMS

As background for the more specific topics dealt with in the following best practice items, this is a summary of some important considerations on how ITS 2.0-supported translation with a content management system can be approached. This is informed by the experiences from a real-life test case that was investigated in MultilingualWeb-LT. Figure 3 gives a general picture of the approach pursued in this context.

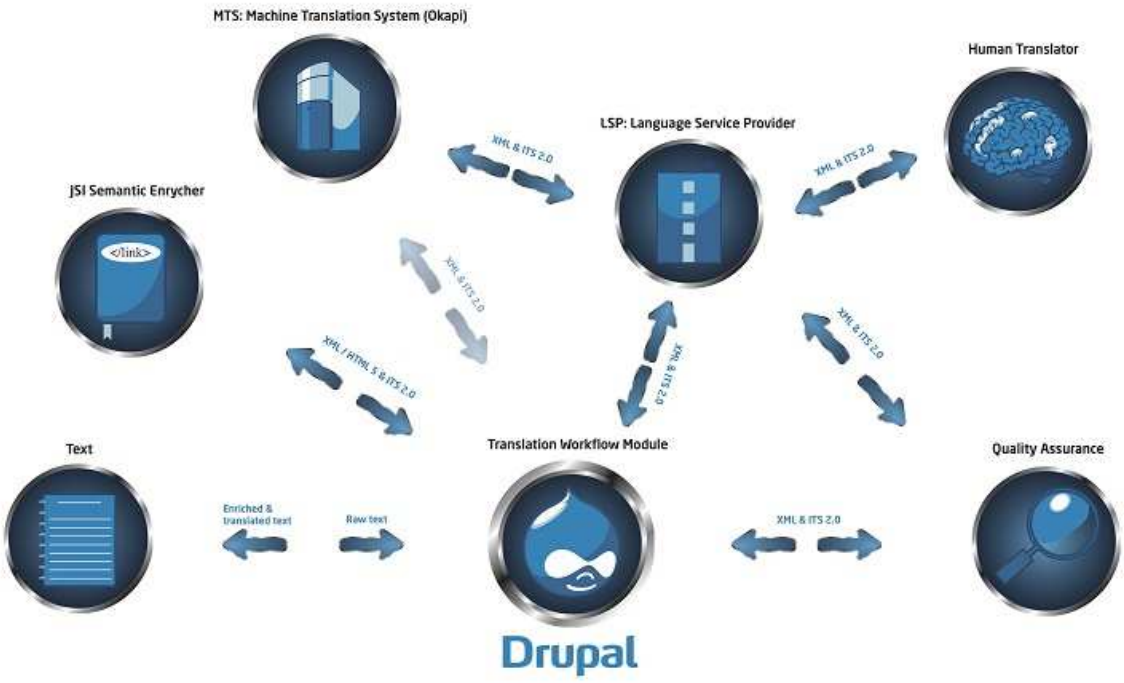

Figure 3: ITS 2.0 within the CMS in MultilingualWeb-LT

1. Before any translation process is started you should think about the general structure of your site and of the content types. Think about if there are any general translations rules you need, like a text which should remain untranslated on all pages or the general domain for all documents. If you have such parts a global rules file should be created and linked from the content pages to it (see the *Global Approach* section of the ITS 2.0 standard). All data categories in ITS 2.0 can be defined with the global approach, but some cannot be defined in a local approach. In this way it is not necessary to repeat local mark-up on all pages. Also it is easier to maintain if changes in the site are happening.
2. Depending on the volume of your page and on how many people are working on it, there are two ways you could define local mark-up. For small sites with only a few people working on them, it should be possible to set ITS 2.0 mark-up directly while the content is created. This means all content creators should be aware what data categories have to be used and how to set these. Bigger sites often have a separate translation manager, who is checking the content and starts the translation process. The translation manager should be able to easily edit and add local ITS 2.0 mark-up in the content, without accidentally changing the content.
3. The translation process itself should be configurable as far as possible. It may involve only one LSP or multiple LSPs and services. The task of the CMS is to send the content to the services in a predefined order. After the user annotates the content other services can be used e.g. to automatically add the text analysis data category, use a machine translation and let it be reviewed by a LSP service (of course the machine translation part might also be covered by the LSP without involving the CMS). In the last step it should be possible to send the translated content to a quality assurance service or review it by yourself.
4. It is important to use the translation process not only for the content on your site, but also for literals like menu titles, error messages, etc.

These considerations went into a solution with the following central characteristics:

- On the content provider's side the creation of the ITS 2.0 metadata aware workflow involves the following areas:
    - Annotation of source language content with ITS 2.0 metadata within the Drupal CMS. Structural annotation rules can be specified as global rules on 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 mark-up. Manual annotation features are available in all generally expected interaction modes (toolbar buttons, context menu, keyboard shortcuts):
    - Two annotation approaches are supported:
        - 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.
        - 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.
    - Transparent data round-tripping: Triggered from within Drupal, this is realized in the background via export/import of files XHMTL+ITS 2.0 mark-up, to be sent to/received from LSP. The process is based on an extended version of the Drupal Translation Management (TMGMT)-module.
    - Translation review: ITS 2.0 mark-up is retained in this step so that annotated information can be taken into account for QA purposes.
- On the LSP side, the creation of the ITS 2.0 metadata aware workflow encompasses three areas:
    - Pre-production/post-production engine for processing content files annotated with ITS 2.0.
    - LSP internal localization workflow to provide support to project management and production

processes.

- Computer Assisted Translation (CAT) tool usage for translation, revision and post-editing with ITS 2.0 annotated content.

4.2. Carefully consider the division of work between content creation and translation

It is important to consider the division of work between content creation and translation because different specific requirements will derive from this decision on the workflow structure. First of all there will be the question whether translation will be done by an external language service provider (LSP) or internally. In the first case it is quite likely that translation will happen inside a system owned by the LSP (e.g. a translation management system), and interfaces of some kind to this system will be needed. In the second case it may be an option to handle translation completely inside the CMS, with no interfacing needed between systems, but with special UI needs to handle the translation work. Second, annotation may be done by the content provider (being the expert needed for content related information), the LSP (as an expert for information needs during translation) or both parties. Third, there may or may not be a review step specific to the question if the translation result conforms to the requirements imposed by the ITS 2.0 mark-up in the source, and such a step may also be conducted by content provider, or partly by both.

4.3. ITS and the review process

The use of ITS introduces a number of additional review tasks to the multilingual content lifecycle. Firstly, the ITS markup itself may be subject to review at certain points. But secondly and more importantly, the information given by the ITS metadata supports a more principled and exact review of the actual translations. Both aspects require some decisions to be taken.

In principle it may make sense to review the validity and adequacy of ITS mark-up at the content creation stage, to check whether the mark-up has been taken into account during the translation, and to assess again the validity and adequacy of ITS mark-up taken over to the target text. All three tasks may be done on the side of the content provider (who will typically have an interest in making sure that translators do not receive any misleading indications, and that they deliver according to expectations), or on the side of the language service provider (who will want to make sure that he receives data that can be processed without further difficulties). The tasks may be performed as separate workflow steps, or together with more generic review activities.

All these decisions will influence to what extent separate tools will be needed to support the review tasks, and whether they should be available within the CMS or in the translation tool suite. The ITS 2.0 integration for Drupal provided in MultilingualWeb-LT provides review facilities to check mark-up in translated text, as well as adherence of the translation to the encoded information, in one step after a translation has been received.