

# ITS2.0 and Computer Assisted Translation Tools

**Pablo Porto, Dave Lewis, Leroy Finn, Christian Saam, John Moran, Anuar Serikov, Alex O'Connor**  
**Centre for Next Generation Localisation**  
**Trinity College Dublin, Ireland**

portovep@tcd.ie, dave.lewis@scss.tcd.ie, finnle@tcd.ie, Christian.Saam@scss.tcd.ie, moranj3@cs.tcd.ie, serikova@tcd.ie, alex.oconnor@scss.tcd.ie

## Abstract

Version 2.0 of the Internationalization Tag Set (ITS) introduces a set of data categories designed to ease the interaction between content management, language technologies and the localization workflow. Many of these data categories capture meta-data related to manual language processing tasks including terminology management, source and target quality assessment and post-editing of machine translation. This paper examines the specific role that ITS2.0 may play in the design of Computer Assisted Translation (CAT) tools. It outlines the requirements ITS2.0 places on CAT tools design. We then examine the implementation of these requirements in a custom ITS/XLIFF-based CAT tool implemented as a client-based web application using JavaScript and compare this to an attempt to implement similar features in an open source Java-based CAT tool.

**Keywords:** *Internationalization Tag Set, Localization, Computer Assisted Translation, ITS 2.0, XLIFF, CAT, Java, Internationalization, Localisation, JavaScript, Requirements, Design, Client-based web application*

## 1. Introduction

Computer Assisted Translation (CAT) tools are a key part of the tool chain that supports the localization workflow. As efficiencies are sought in the localization process, the design of CAT tools is becoming an increasingly important area for investigation, especially in relation to how it supports integration with automated language technology components such as machine translation, terminology extraction and quality assessment. The MultilingualWeb-Language Technology (MLW-LT) working group at the W3C has recently finished specifying the successor to the Internationalization Tag Set (ITS) 1.0 (Lieske & Sasaki 2007) in the form of the ITS2.0 specification (Filip et al 2013). This extends the scope of ITS beyond the internationalization concerns of version 1.0 and addresses the exchange of content meta-data across the localization workflow, encompassing the CAT tool. It specifically addresses meta-data intended to assist in the integration of language technologies into the localization workflow, and therefore includes meta-data that should be displayed, created and manipulated via CAT tools. This paper examines the use cases in which CAT tool users may interact with different ITS2.0 data categories. We then review two exploratory implementations of these use cases. First, in detail, we examine the implementation of a green-field CAT tool implementation as a web client application using JavaScript. Then we briefly review

an attempt to introduce ITS features into an existing open source CAT tool implementation, OmegaT<sup>1</sup>.

## 2. CAT tool Use Cases for ITS2.0

This section outlines requirements for supporting data categories from ITS2.0 that are relevant to the operation of CAT tools. We assume ITS2.0 meta-data both consumed and generated by a CAT tool will be accessed and stored as part of an XLIFF 1.2 file (Savourel et al 2008). Therefore these requirements aim to align with the work on ITS-XLIFF mapping<sup>2</sup> being undertaken by the ITS Interest Group.

The ITS standard associates meta-data, in a standard format, with both source and target text. A number of text annotation meta-data types, called data categories, are defined for ITS2.0, including support for ones defined in ITS1.0. Figure 1 summarises the set of data categories, with the ones relevant to CAT tool design underlined. With reference to XLIFF concepts, where content is presented to translators as individual pairings of source and target segments, ITS mark-up may be presented to a CAT tool user in association with the following:

- source segments or sub-segments
- suggested target segments or sub-segments (e.g. taken from the XLIFF 'alt-trans' elements)
- target segments or sub-segments based on

suggested target selection and post-editing by the tool user

- target segments or sub-segments provided in the XLIFF file and being reviewed or revised by the tool user.

response to a corresponding sub-segment protected section in the source. Note this behavior is not specified for the translate data category in the standard but seems a useful feature.

ITS1.0	I18n	Language Technology	Provenance & QA
<ul style="list-style-type: none"> <li>• <u>Translate</u></li> <li>• <u>Localization Note</u></li> <li>• <u>Terminology</u></li> <li>• Directionality</li> <li>• Ruby</li> <li>• Lang info</li> <li>• Element within text</li> </ul>	<ul style="list-style-type: none"> <li>• Locale Filter</li> <li>• External Resource</li> <li>• Preserve Space</li> <li>• <u>Allowed Characters</u></li> <li>• <u>Storage Size</u></li> <li>• ID Value</li> </ul>	<ul style="list-style-type: none"> <li>• <u>Domain</u></li> <li>• <u>MT confidence</u></li> <li>• <u>Text Analysis</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Quality Issue</u></li> <li>• <u>Quality Rating</u></li> <li>• <u>Provenance</u></li> </ul>

Figure 1: ITS Data Categories with CAT tool relevant ones highlighted

Use cases are given below on individual ITS data categories consistent with the ITS notion that systems can conform to each data category independently of support for the other. However, any support for multiple data categories in CAT tools must also support their concurrent usage. So it is important that all visual indications at the segment or sub-segment level should be visually distinct from each other. Care is needed however to ensure that CAT tools users do not suffer cognitive overload due to any proliferation of displayed meta-data associated with the text being translated. The following use cases are listed against the relevant ITS2.0 Data Category. Reference is made to the relevant ITS markup indicated by prefix “its:” and the XLIFF mark-up by the prefix “xlf”.

**Translate**

This data category indicates whether the annotated text should be translated or not:

- **TraUC1:** View segments marked not to be translated (using its:translate=”no”) as context to the CAT user
- **TraUC2:** View highlighted source sub-segment that are marked to be not translated (using xlf:mrk mtype= “protected” as the equivalent of as its:translate=”no” per the MLW-LT ITS-XLIFF mapping) to guide segment translation.
- **TraUC3:** View a highlighted sub-segment of a suggested target translation marked to indicate where an MT engine has specifically not translated text in the xlf:alt-trans/target in

**Localization Note**

This provides a way to convey a note from content authors or other down-stream internationalization workers to workers in the localization workflow.

- **LocUC1:** View the note text and note type (description or alert) associated with source segments or sub-segments marked.

**Terminology**

This indicates if the annotated text constitutes a term and references associate meta-data)

- **TrmUC1:** View source sub-segments annotated as terms (with its:term=”yes”) together with, where present, a clickable link to further information on the terms (from its:termInfoRef); the confidence score associate with this term identification (from its:termConfidence) and a clickable reference to more information on the tool that generated this meta-data (from its:annotatorsRef).
- **TrmUC2:** Create source sub-segment term annotations, together with: an optional reference to further information (populating its:termInfoRef) and an optional manually determined score of the users confidence in the term annotation (populating its:termConfidence). If the latter is added, then the corresponding its:annotatorsRef attribute must be added by the CAT tool. This should reference the CAT tool itself, but could also usefully provide information about the user. If the schema of the referenced terminology

information resource is known, the values could be pre-fetched and displayed instead of presenting the references.

- **TrmUC3:** Existing source sub-segment term annotations may be deleted by the tool user.
- **TrmUC4:** Edit existing source sub-segments term annotations. This may involve changing or adding the value of reference additional information (modifying its:termInfoRef). It may also involve the changing the its:term value (“yes” or “no”) or changing the its:termConfidence according to the terminology procedure being followed by the user in checking and correcting. If present the its:annotatorsRef value cannot be changed by the tool user.
- **TrmUC5:** View term annotation of a target language sub-segment. This indicates that an automated translation component has either attempted to preserve terminology annotation of the corresponding source segment or has added the annotation based on internal terminology information. This can be useful in assuring target terminology quality and consistency.

#### Domain

This indicates the application domain addressed by the annotated text.

- **DomUC1:** View domain annotations associated with the entire source document, a specific subsection of it, segments or sub-segment (as represented by annotations of xlf:file, xlf:trans-unit, xlf:source-segment or xlf:mrk respectively by its:domains). Differences in domain annotation that is different from surrounding text should be differentially highlighted. Note that the value of the its:domains attribute can be multivalued.
- **DomUC2:** View an automatically generated translation (i.e. the xlf:target in an xlf:alt-trans) that has to be annotated with its:domains to indicate the actual domain values used in the translation. This may be important if the value used by the MT engine differs from those specified in the corresponding source segment or sub-segment.

#### Text Analysis

This annotates text with reference to lexical or semantic information.

- **TxaUC1:** View source sub-segments annotated with text analysis. Where present, the clickable values of the class of entity the text represents

(from taClassRef) and the specific instance it represents (from its:taSource and its:taIdent or its:taIdentRef). If the schema of the reference resources is known, the values could be pre-fetched and displayed instead of presenting the references. Also, if a score of the confidence in the annotation is present (from its:taconfidence) this should be presented together with a clickable reference to more information on the tool that generated this meta-data (from its:annotatorsRef).

- **TxaUC2:** Create source sub-segment text analytics annotations. In such cases, if an its:taConfidence score is added then the corresponding its:annotatorsRef should identify the CAT tool and possibly also the user.
- **TxaUC3:** Delete existing source sub-segment text analysis annotations.
- **TxaUC4:** Edit existing source sub-segments text analysis annotations. This may involve changing or adding the value of its:taClassRef, its:taSource and its:taIdent or its:taIdentRef attributes. It may also involve changing the value of the its:taConfidence attribute according to the text analysis processing procedure being followed by the user in checking and correcting. If present, the value of the its:annotatorsRef attribute cannot be changed by the tool user.
- **TxaUC5:** View target language sub-segments that have been annotated with text analysis annotation. This can be used to indicate that an automated translation has either attempted to preserve source text analysis annotation of the corresponding source segment or adds the annotation based on text analysis functionality integrated with translation workflows. This can be useful in supporting target terminology consistency.

#### MT Confidence

This provides a confidence score resulting from an automated translation of the annotated text)

- **MtcUC1:** View the confidence score of a machine translation represented at a suggested target segment level by an xlf:alt-trans/target (from its:mtconfidence) and a clickable reference to the MT engine that produced the annotation (from its:annotatorsRef).
- **MtcUC2:** If a post-editor selects an xlf:alt-trans/target element as the translation of the corresponding source segment such that it is replicated in the xlf:trans-unit/target element and if that translation remains unaltered (i.e. it is not post-edited) then that element should be

annotated with `its:mtConfidence` and `its:annotatorsRef` attributes from the corresponding `xlif:alt-trans/target` element.

- **MtcUC3:** In situations where sub-segments have a differential MT confidence (whether in an `xlif:alt-trans` and the `xlif:target` element) this need to be visually indicated to the tool user. If the differential sub-segment confidence score is the result of translation by different engines, then the corresponding different engine annotation (`its:annotatorsRef` attributes) should be used. Note, that sub-segment confidence score are not currently supported in the ITS-XLIFF mapping []

### Provenance

This records the people, tools and/or organizations involved in translating or revising the translation of the annotated text.

- **PrvUC1:** View translation provenance annotation applied to target segment and suggested target segments, displaying the value of the tool, organization and person involved if present (from `its:tool`, `its:org` and `its:person` attributes) or presenting clickable links for the same (from `its:toolRef`, `its:orgRef` and `its:personRef`). in a way that the user can opt to retrieve the referenced information. Similarly, view translation revision annotation associated with target segments that have undergone post-editing. In both cases multiple records may apply, so the display of attributes must indicate their grouping into individual records.
- **PrvUC2:** View a clickable reference to further provenance information (from `its:provRef`) if present. Where the tool to be able to determine the type of information being referenced, view it directly in an appropriate format, e.g. W3C provenance format or iOmegaT transLog post-editing logs.
- **PrvUC3:** For each translation or post-editing session, populate translation or translation revision provenance information for the segments being addressed in the session and optionally provide a UUID value for the `its:provRef` attribute. If the tool, organization and person values are identical to an existing record, then the same record reference should be used, but a new UUID should be appended to the value of the `its:provRef` attribute.

### Localization Quality Issue

This records a encoding of a quality assessment applied to either source or target text.

- **LqiUC1:** View localization quality records annotating any source or target segments and any source or target sub-segments. Each record may include a type string, some comment text, a severity value (between 0-100), a profile reference that can be clicked to display details of the localization quality reference schema used and an flag indicating whether the issue is currently in active or not. These are taken, respectively, from:
  - `its:locQualityIssueType`,
  - `its:locQualityIssueComment`,
  - `its:locQualityIssueSeverity`,
  - `its:locQualityIssueProfileRef`
  - `its:locQualityIssueEnabled`.
- **LqiUC2:** Add new localization quality issue annotations to either source or target segments or to source or target sub-segments.
- **LqiUC3:** Edit existing localization quality issue annotations to correct errors they made in previous annotations. Changes to annotations provided by previous users should be restricted according to localization quality checking procedures, including changing the status of the issue enabled flag.
- **LqiUC4:** Delete an existing localization quality issue annotation to correct erroneous annotation they made previously. Deletion of annotations provided by previous users should be restricted according to localization quality checking procedures.

### Localization Quality Rating

This allows annotation of an overall quality rating for a target document or section or of a quality vote for a particular document, section, segment or sub-segment (including suggested segments and sub-segments).

- **LqrUC2:** View the meta-data associated with the annotation, namely
  - `its:locQualityRatingScore`,
  - `its:locQualityRatingScoreThreshold`,
  - `its:locQualityRatingVote`,
  - `its:locQualityRatingVoteThreshold`
  - `its:locQualityRatingProfileRef`.
- **LqrUC2:** Annotate the whole document, a translation unit, a segment or a sub-segment with a localization quality rating as a score or as a vote. For specifying a vote, some external mechanism is required for tallying the vote. The option should be offered for the user to enter a threshold value for the rating or vote and a reference URL to the assessment framework

used.

- **LqrUC3:** Delete an existing annotation.

**Storage Size**

This specifies the maximum storage size allocated to the annotated content.

- **StsUC1:** View storage size information for a target segment or sub-segment (from its:storageSize, its:storageEncoding and its:lineBreakType).
- **StsUC2:** Annotate a target segment or sub-segment with storage size information, indicating size restrictions on the textual content (populating its:storageSize, its:storageEncoding and its:lineBreakType).
- **StsUC2:** View report on breaches of the storage size restriction of the annotated textual.
- **StsUC3:** View proportion of the allowable storage size restriction available on the annotated textual content as it is being edited and be altered when the maximum is reached.

**Allowed Characters**

Specifies the characters that are permitted in a given piece of content.

- **AlcUC1:** Annotate a target segment or sub-segment with an its:allowedCharacters attribute to indicate which characters are permitted in the textual content.
- **AlcUC2:** Be alerted where target or suggested target text (from xlf:trans-unit/target and xlf:alt-trans/target) conflicts its:allowedCharacters value, indicating which characters in the text are in conflict.

**3. Implementing ITS/XLIFF based CAT tool as a Web Client Application**

In this section we describe an initial proof of concept implementation of these requirements that was implemented as a Web Client application using Java Script such that stand alone CAT tool functionality could be offered in a web browser. This was in part an assessment of the level to which a CAT tool based on ITS and XLIFF standard could be built using the Open Web Platform<sup>3</sup>. To put this in context, Table 1 summarizes the level to which the features of ITS2.0 and XLIFF integration are supported by equivalent features offered by other existing CAT tools.

The application, named Escriba, needed to retrieve and store both the localization content and the ITS meta-data associated with it, so an ITS parser is

	Wordfast Anywhere	Google Translator Toolkit	Pootle	XTM Cloud	PO Editor	Microsoft Translator Hub
XLIFF Support	N	N	Y	Y	N	N
ITS Integration	N	N	N	Y v1.0	N	N
Web MT	Y	Y	Y	Y	Y	Y
Glossary/Termbase	Y	Y	Y	Y	Y	Y
Integrated Spell Checker	N	Y	N	Y	N	unknown
Project Management features	N	Y	Y	Y	Y	Y
Project statistics	Y	Y	Y	Y	Y	Y

**Table 1:** Feature comparison of existing web client CAT tools

required. An existing parser implemented in jQuery was considered. This was called, jQuery ITS2.0 Parser<sup>4</sup> and is developed and maintained by Cocomore, one of the active members in the MLW-LT WG. However this library worked with XHTML, so for this CAT tool implementation, its use would require a conversion from XLIFF to XHTML and back again. Initial feasibility testing show this to be less efficient that developing a single ITS2.0+XLIFF parser, so this latter option was adopted.

**Implementation Components**

The overall design consisted of the following modules and constituent components:

**ITS2 Module:**

This encapsulates components which provide support to view, edit and delete ITS 2.0 meta-data. To date implementation supports the Translation, Localization Note, MT Confidence, Provenance and Localization Quality Issue data categories. As per the ITS Interest Group’s ITS-XLIFF mapping only the local style of ITS annotation is supported, i.e. global selector style was not supported. This module is formed of the following components:

- its-metadata-editor: Allows insertion, edition and deletion of ITS 2.0 meta-data in a given XLIFF file.

- its-metadata-visualizator: Contains all the logic which specifies how the information extracted from the ITS 2.0 metadata should be displayed.
- its-metadata-extractor: Provides the required functionality to extract ITS 2.0 from a given XLIFF file

#### **XLIFF module:**

This module contains all the components that provide support for handling XLIFF files. It is formed of the following components:

- xliiff-data-manipulator: Allows for insertion, editing and deletion of the XLIFF elements of a given XLIFF file.
- xliiff-data-selector: Provides support for selecting specific XLIFF elements (e.g. target elements) of a given XLIFF file.

#### **Core module**

This module contains the core functionality of the system. It is formed of the following components:

- content-navigation: Controls how the content of a project file should be displayed and in what order. It contains almost all the User Interface (UI) functionality.
- core: Provides support for down-loading and uploading XLIFF files and the functionality for set up a new project.

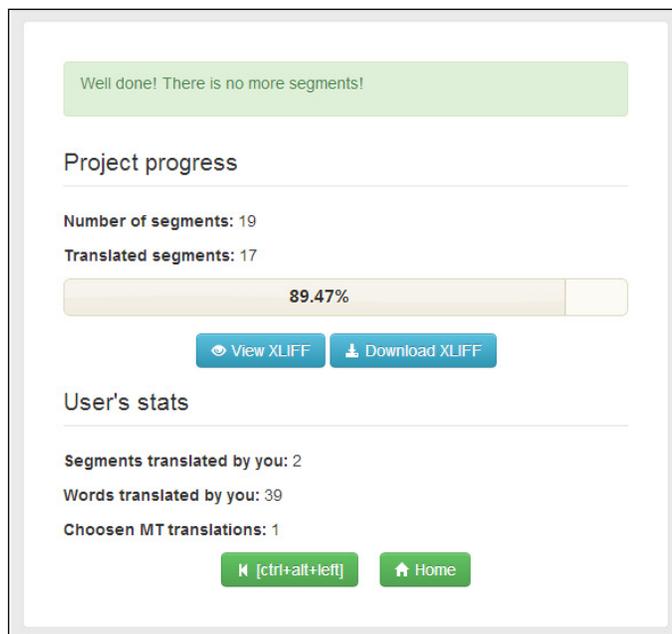
It also contains the remaining UI functionality which is not defined in the content-navigation module.

- user-pref-controller: Allows the configuration of preferences for a specific user. It stores and retrieves user preference information in a user model.
- keyboard-shortcuts: Controls the keyboard shortcuts supported by the system and how to enable or disable it.

#### **User Interface Implementation**

The implementation of the user interface (UI) is one of the key aspects of the project. It is necessary to select the right set of technologies which allow the implementation of a clear and appealing interface to engage users. It is also designed to enable Adaptivity in the UI so that various features can be presented in different ways for different types of users. There are several UI front-end frameworks which ease the design and implementation of a richer user experience (UX). Such frameworks collect best practices and UI and UX conventions and bundle for used by developers who are less expert in the UI/UX area. The one selected for this project was Bootstrap<sup>5</sup>, which was released by Twitter as an open source UI front-end to provide a simple and quick way of creating clean and highly usable applications.

We have used Bootstrap version 2.3.2 to quickly develop the basic functionality of the UI of the



**Figure 2:** Bootstrap component example - Escriba progress Display

system. Moreover, one of the benefits of implementing the presentation layer using Bootstrap is that this framework integrates a set of responsive features that will allow the prototype to be adapted to mobile phones very quickly in the future. Figure 2 shows an example of how different elements of the Bootstrap toolkit were used to provide an appealing and simple UI for progress tracking. Bootstrap aids this through high level provision of different types of buttons, icons and typography. However UX is about more than a clean and usable user interface. The UI should react quickly to the user interaction and provide visual clues that this interaction is happening like component animations, transitions, effects, etc. The Escriba implementation achieves some of these effects using jQuery<sup>6</sup>. This is an open source library developed by the UX community to ease the implementation of highly interactive web applications. Figure 3 shows an example of how the jQuery UI accordion widget was used to present the user with different alternative translations for a source segment using an animated expanding widget. The combination of Bootstrap and jQuery UI have allowed us to quickly achieve an acceptable UI and UX design for the Escriba prototype match the needs of CAT tool users interacting with ITS meta-data.

retrieve specific nodes of the XLIFF document. This was selected over XPath due to the apparent simplicity and familiarity of using these CSS selectors for styling purposes in HTML pages. The CSS selectors worked fine for all usage scenarios except when selecting an element based on an attribute whose name contained a colon, though a work around was achieved.

### Main UI Features

We now describe the main feature of the Escriba UI. A demonstration version of Escriba is available on the web<sup>7</sup>.

### Home page

The main page of the web application allows the user to create a new project by uploading a XLIFF file. As can be seen in Figure 4, this page is also used to provide access to the different tasks that make up the provided CAT tool functionality as well as a list of XLIFF+ITS2.0 input samples to see the implemented ITS 2.0 support in action.

### Translation panel

The translation panel (see figure 5) is the main component of the application. It allows the user to translate segments, edit translations and see

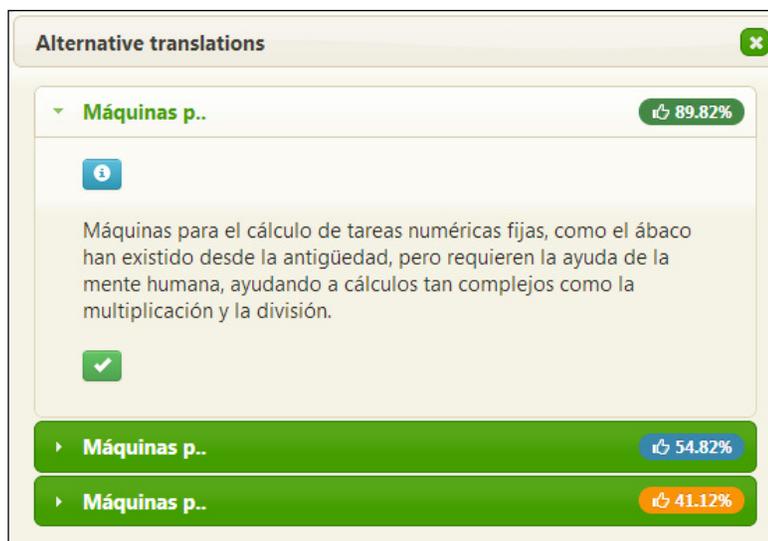


Figure 3: jQuery accordion example for revealing alternative suggested translations

### XLIFF Manipulation

As detailed earlier a dedicated XLIFF parser was developed to efficiently support import and export into the UI components. There are several query languages that can be used for selecting nodes of an XML document using JavaScript. We have developed a custom parser that uses CSS selectors to

alternative translations for a given segment. Moreover, it allows the user to access to more advance features related to ITS meta-data creation, edition and deletion process. The segments are presented in a vertical list. Segment number, source text and target text are shown for each of the segments. The user can select a segment by clicking

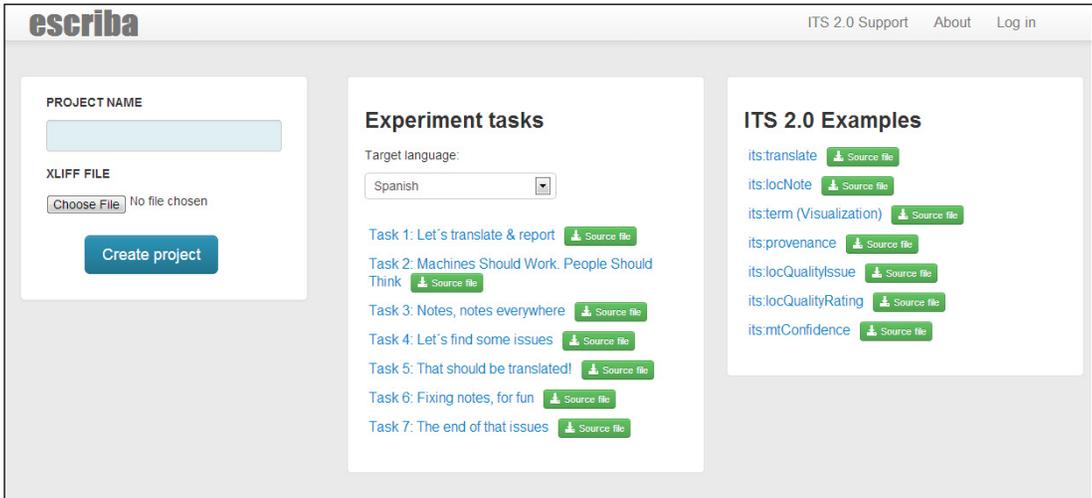


Figure 4: Escribe Demo Version Home Page

on it and see which segment is selected at any time. The ITS 2.0 information associated with a given segment is shown above the source text or target depending on the one with which it is associated. The user can create new ITS 2.0 annotations for Localization Quality Issues and Localization Quality Rating through the buttons situated below the source and target text. Finally, two buttons can be found at the bottom of the page which allow forward and backward navigation through the segment list.

**Progress panel**

The progress panel (see Figure 2) can be accessed

from the top of the translation panel. It shows project related information such the project name as well as the current progress and the user performance in the current project. The progress is represented by a percentage and the user can also see the total number of segments that the project includes and the number of already translated segments. With regards to the user statistics, the user can see how many segments were translated by him, the number of words translated and finally the number of alternative translations selected.

As can be seen in Figure 2, the panel also allows the

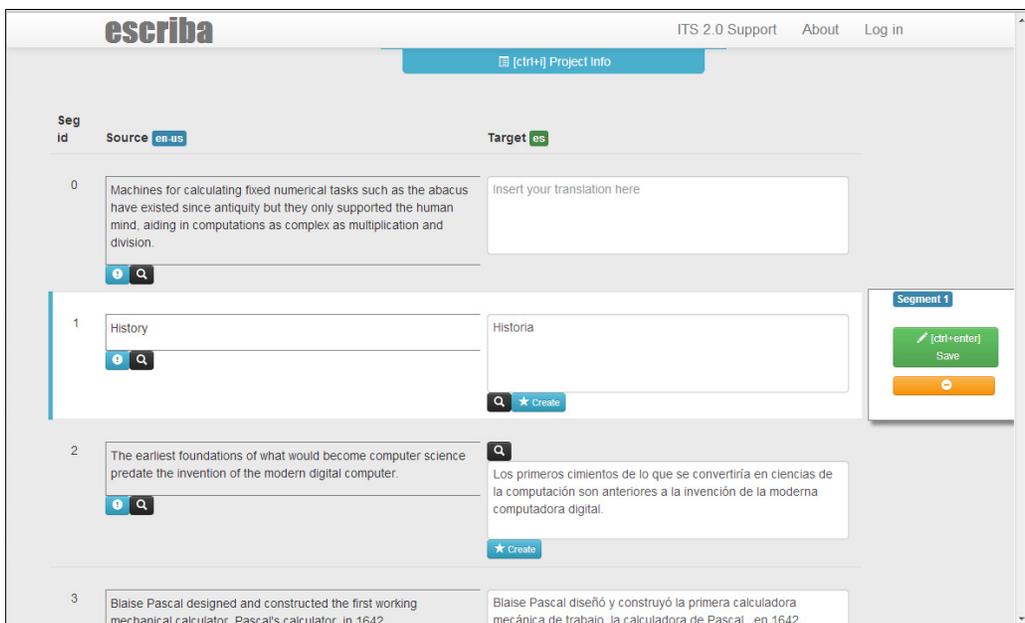


Figure 5: The Translation Panel

user to download or view the current state of the XLIFF file associated with the project.

#### 4. Alternative Implementation

An internal document data structure, that reflects the structure of the source/target document well and in a manner agnostic of specific document type, greatly eases some aspects of rendering and manipulating the contained meta-data. As we have seen in the implementation of Escriba, ITS lends itself well to W3C DOM data structures and in turn to simple implementation using widely available open source web application libraries in JavaScript.

As a comparison, we also investigated the handling of ITS enriched translation interchange documents in a Java-based open source CAT tool called OmegaT. The Swing GUI toolkit used in this application provides a rich graphical editor API. The backing data model of the component representing the main editor is based on a hierarchical (tag) structure inspired by HTML/XML and thus similar in spirit to the W3C DOM. In principle this allows one to easily represent any kind of document structure/type with or without (in-line) ITS meta-information. In the data model the structure and content are held separately. The same content may even be associated with several different structures representing different aspects. Although OmegaT's editor interface is based on the general swing editing components, the part of the editor's backing document model representation structure is basically circumvented in favor of a secondary data structure that represents the source document's content as well as those parts of its structure deemed necessary for achieving the purpose of translation. This secondary data structure is a list containing the equivalents of segments and is kept in sync with the editor's content model. The list is populated when the source document is read in. Document filters analyse input documents in terms of content-bearing (i.e. translatable) and non-content-bearing structural elements. Content is extracted from the content bearing elements and the general structure is copied into a backup document skeleton that is not accessible from the editor. Thus upon import a lot of the structural information is lost to further manipulation along with any meta information embedded at that level.

In undertaking this alternative implementation it became clear that the extent of changes needed for a full information round-trip from document to editor and back required a much greater the time frame and

skill level compared to the green-field implementation of Escriba. For some aspects of ITS that can be expressed by features of XLIFF (such as MT confidence, non-translate segments and localization notes) and that survive the existing input filtering process, implementations of GUI representations and interaction patterns were attempted. For clarity of presentation, several GUI components including the main editing component were changed from a line oriented to a tabulated display. To visualise these unsupported features we prototyped a number of tabular layouts for ITS 2.0 segment-level information and terminology information in the GUI, though these additional meta-data were not supported by existing data import filter.

#### 5. Conclusions

ITS2.0 defines a set of meta-data that can be associated with content as it passes through various stages of the localisation process. Many of these data categories therefore must be viewed and manipulated by translators using CAT tools. In this paper we explore the requirements that CAT tool developers need to satisfy in order to support such interactions. We also conduct some implementation trials. As may be expected, support for these new meta-data features was found to be much easier in a green field implementation than when trying to refactor an established CAT tool. However, the Escriba implementation experience also showed the potential for rapid development of flexible and engaging CAT UI function that operates in a web browser by leveraging modern Open Web Platform libraries such as Bootstrap and JQuery. The Escriba implementation was developed by a master's student in a few weeks, and was then easily extended to include adaptive UI features. The OmegaT implementation was also able to support efficient UI prototyping using Java Swing, but was restricted by the complexity of refactoring the import and export feature to accommodate new meta-data types. If the integration of new technology such as machine translation, text analytics and quality assessment, is to be integrated into the localization workflow using meta-data annotations as advocated by the W3C in ITS 2.0, then such filters must accommodate more flexible means for accommodating new meta-data.

#### Acknowledgements

This research is supported by the European Commission as part of the MultilingualWeb-LT

project (contract number 287815) and by the Science Foundation Ireland (Grant 12/CE/I2267) as part of the CNGL – the Centre for Global Intelligent Content ([www.cngl.ie](http://www.cngl.ie)) at Trinity College Dublin and under Grant 12/TIDA/I2424. The authors would like to thank Sharon O’Brien of CNGL at DCU as well as the members of the MLT-LT WG for feedback provided on initial drafts of ITS-CAT tool requirements.

## References

- Filip, D., McCance, S., Lewis, D., Lieske, C., Lommel, A., Kosek, J., Sasaki, F., Savourel, Y. (2013) ‘*Internationalization Tag Set (ITS) Version 2.0*’, W3C Recommendation, available <http://www.w3.org/TR/its20/> [accessed 29 October 2013]
- Lieske, C., Sasaki, F., 2007, *Internationalization Tag Set (ITS) Version 1.0*, W3C Recommendation available: <http://www.w3.org/TR/its/> [accessed 29 October 2013]
- Savourel, Y., Reid, J., Jewtushenko, T., Raya, R.M., (2008), *XLIFF Version 1.2.*, OASIS Standard available: <http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html> [accessed 29 October 2013]

## Notes

- <sup>1</sup> <http://www.omegat.org/>  
<sup>2</sup> [http://www.w3.org/International/its/wiki/XLIFF\\_1.2\\_Mapping](http://www.w3.org/International/its/wiki/XLIFF_1.2_Mapping)  
<sup>3</sup> [http://www.w3.org/wiki/Open\\_Web\\_Platform](http://www.w3.org/wiki/Open_Web_Platform)  
<sup>4</sup> <http://plugins.jquery.com/its-parser/>  
<sup>5</sup> <http://getbootstrap.com/>  
<sup>6</sup> <http://jquery.com/>  
<sup>7</sup> <http://4.mobile-weecat.appspot.com/>