



D3.2.1: OKAPI OCELOT

Phil Ritchie

Distribution: Consortium Internal Report

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	3.2.1
Deliverable title:	Report on VistaTEC deliverables
Dissemination level:	CO
Contractual date of delivery:	31 st March 2012
Actual date of delivery:	08/12/2013
Author(s):	Phil Ritchie, CTO, VistaTEC
Participants:	UL, DFKI
Internal Reviewer:	TCD
Workpackage:	WP1
Task Responsible:	@@@
Workpackage Leader:	Felix Sasaki

Revision History

Revision	Date	Author	Organization	Description
1	11/09/2013	Phil Ritchie	VistaTEC	Initial version.
2	08/12/2013	Phil Ritchie	VistaTEC	Final version.

CONTENTS

Document Information	2
Revision History	2
Contents	3
1. Executive Summary	4
2. Reviewer's Workbench Features	4
3. Reviewer's Workbench Benefits	4
4. Reviewer's Workbench User Interface	5

1. EXECUTIVE SUMMARY

VistaTEC's primary output from this project is a platform independent desktop editor called "Ocelot". The editor can read/write xliiff+its files; is based on the Okapi framework and thus integrates seamlessly with the Okapi workflow.

Ocelot implements the Language Quality, MT Confidence and Provenance data categories of ITS 2.0. Metadata from these categories that is attached to translation units in the XLIFF file can be rendered alongside those segments in a user-definable way through a mechanism of rendering rules. In addition to rendering the metadata, the rules can be used to filter segments according to values of the metadata.

The benefit of a translator or post-editor being able to see this metadata within their editing environment is that their task can be directed and made more efficient: hints about errors and information of a segment's provenance can aid the translator in deciding how to edit a segment, if at all.

Ocelot is written in Java which makes it platform independent allowing for maximum deployment. See "*Ocelot Open Source And Technical Details*".

2. OCELOT FEATURES

Ocelot has the following features:

- Reads/writes Language Quality Issue and Provenance ITS 2.0 data categories. Also reads MT Confidence data category.
- User configurable rules for defining how metadata is rendered in the sidebar alongside each segment.
- User configurable rules for defining how segments can be filtered according to properties of the metadata.
- User configurable keyboard shortcuts for adding new Language Quality Issue metadata.

3. OCELOT BENEFITS

Ocelot increases the efficiency and accuracy of the linguistic review and post-editing processes by providing translators with access to useful contextual metadata that they may not otherwise have. Metadata gathered at earlier stages of the localization process, such as, automated translation; automated quality assurance checks; NLP-based analytical processes; etc. can inform and direct the translator as to actions that they might want to take.

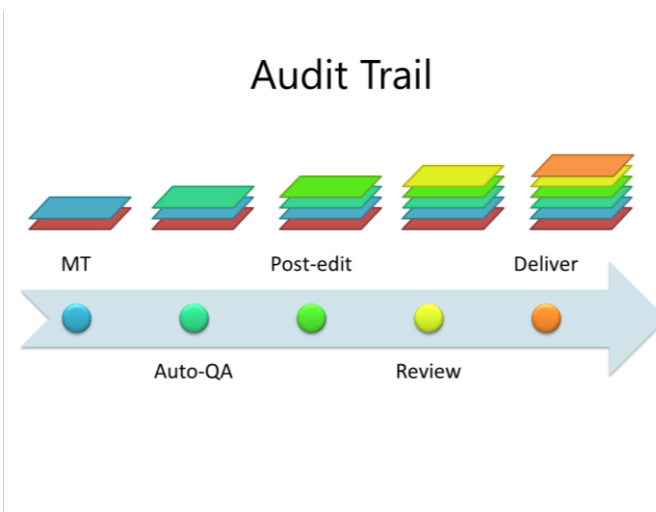
By way of a concrete example:

A document may be translated partially using machine translation and human. At this stage of the process the following Provenance metadata could be gathered:

- Machine translation output confidence scores,
- Name and version of the machine translation engine,
- Name of the human translator and the organization they work for.

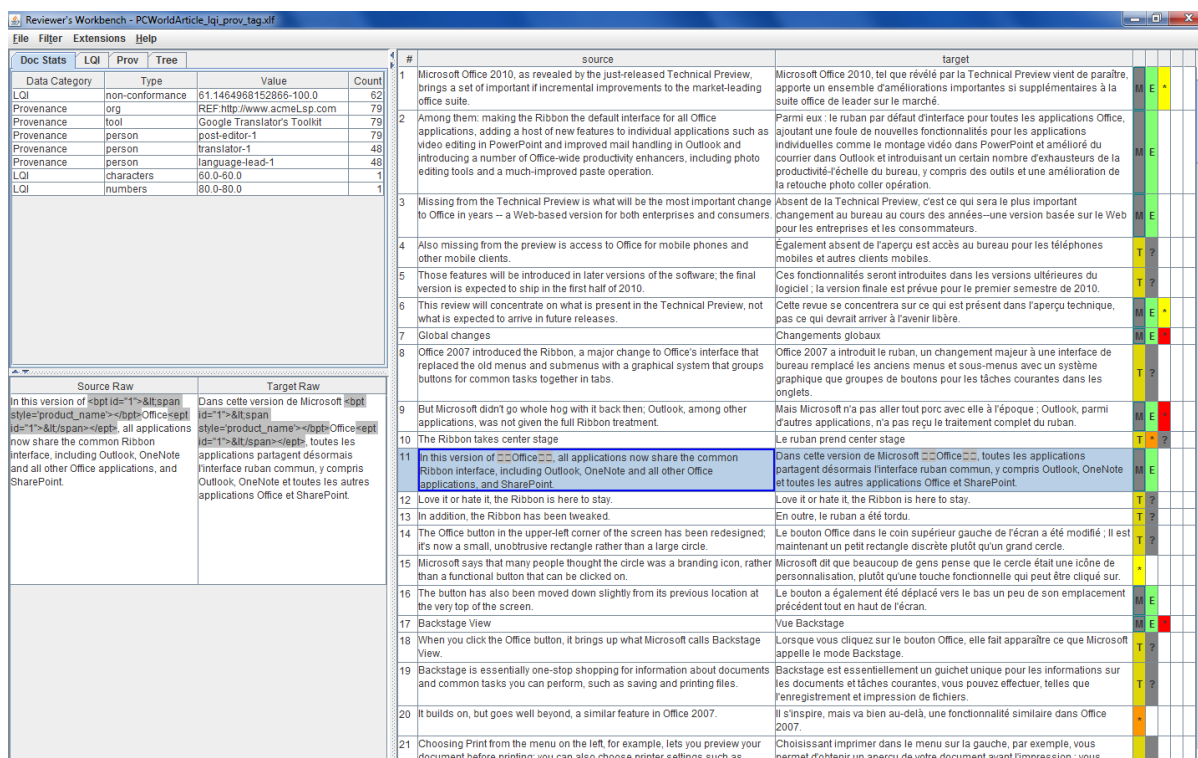
Following translation the document could be passed through an automated quality assurance pipeline. During this phase of the localization process the metadata listed below could be added to the segments:

- Name and version of the automated QA programs,
- Types and severity of errors found,
- NLP/Text Classification hypotheses as to the quality and suitability of the translations.



Thus when the document arrives at the desk of a linguistic reviewer, all of this information is at their finger tips. This can enable them to make decisions as to the strategy of their task: address errors first according to severity level, review only those segments proposed by machine translation, specifically review segments proposed by a translator who is known to be a novice, etc.

4. OCELOT USER INTERFACE



The screenshot shows the 'Reviewer's Workbench' window. The top-left panel displays a summary of metadata:

Data Category	LQI	Prov	Type	Value	Count
LQI			non-conformance	61.1484968152866-100.0	62
Provenance		org		REF: http://www.acmel.ssp.com	79
Provenance		tool		Google Translator's Toolkit	79
Provenance		person	post-editor-1		48
Provenance		person	translator-1		48
Provenance		person	language-lead-1		48
LQI			characters	60.0-60.0	1
LQI			numbers	60.0-60.0	1

The main window displays a comparison between source and target text. The source text is in English, and the target text is in French. The interface includes a 'Doc Stats' panel, a 'Source Raw' panel, and a 'Target Raw' panel. The main window shows a list of segments with their respective source and target text, and a color-coded status column (M, E, T, ?).

Figure 1: Main Workbench window. Top left-hand panel shows summary of all metadata found in the file. Bottom left-hand panel shows segment with raw mark-up. Left-hand side of main window shows metadata rendered according to user-defined rules.

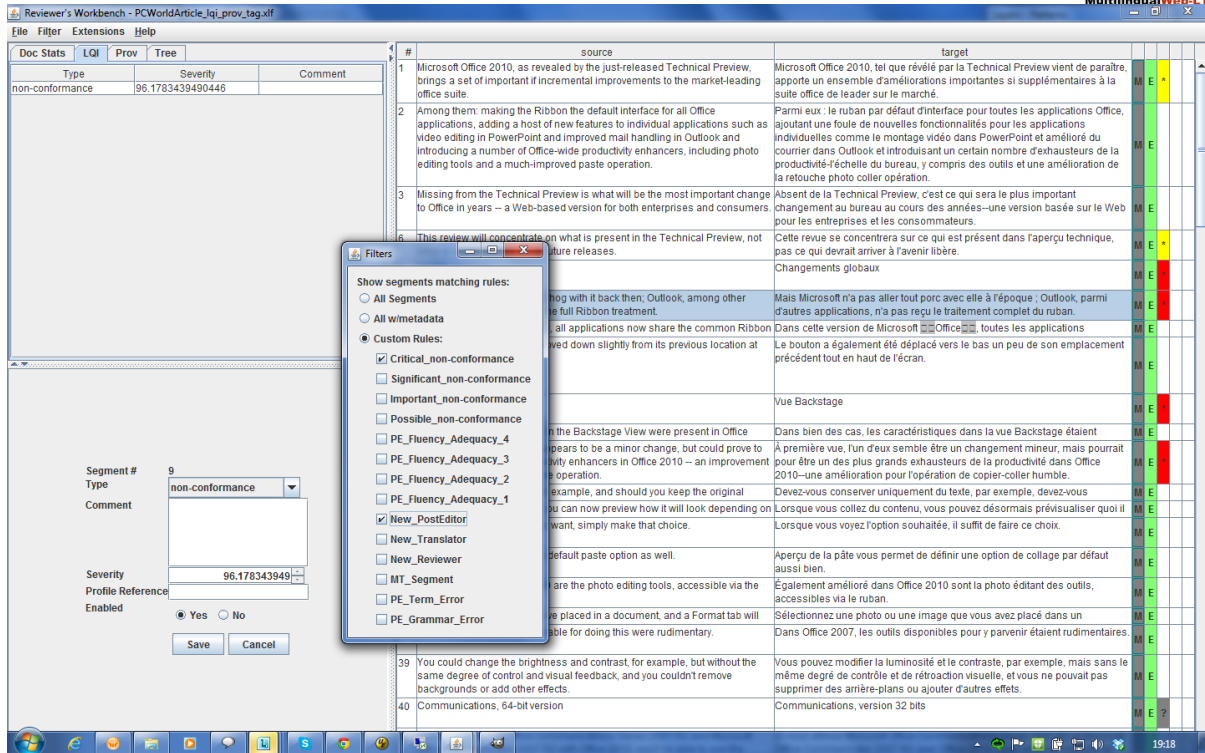


Figure 2: Popup window shows rules associated with particular metadata characteristics used as filtering rules.

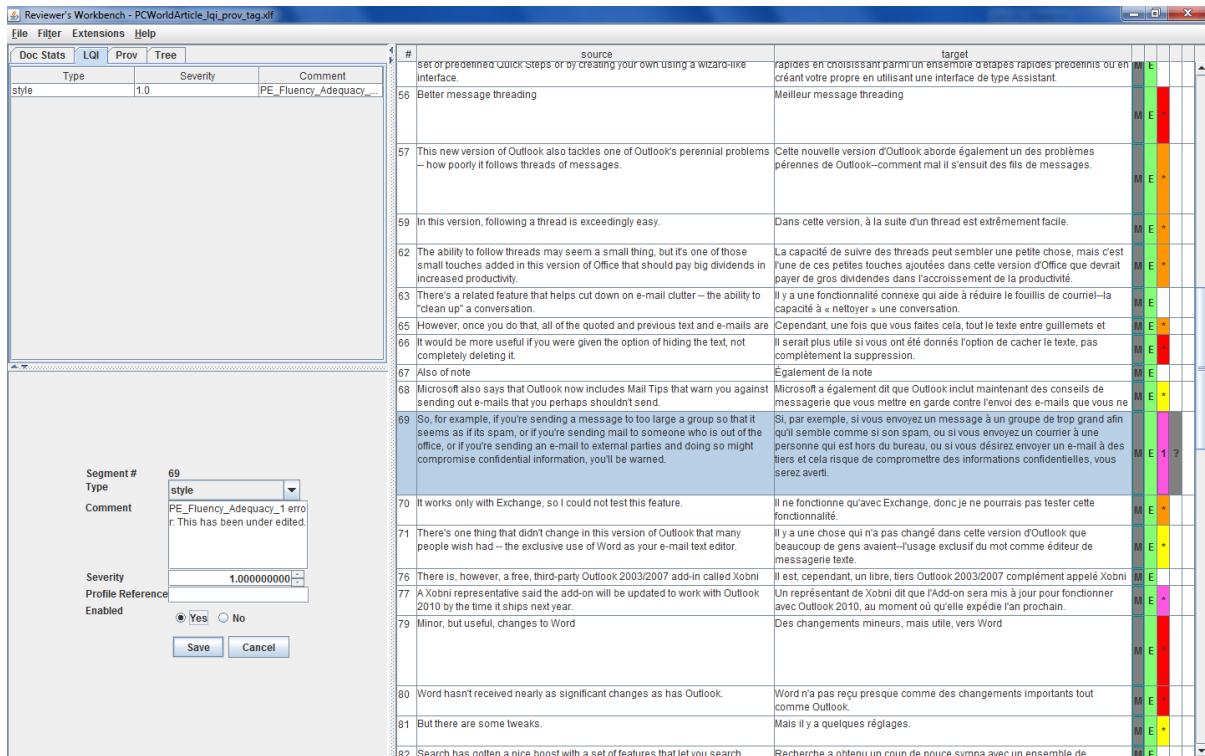


Figure 3: Bottom left-hand panel displays the Language Quality Issue data category values associated with the selected segment in the main editor window.

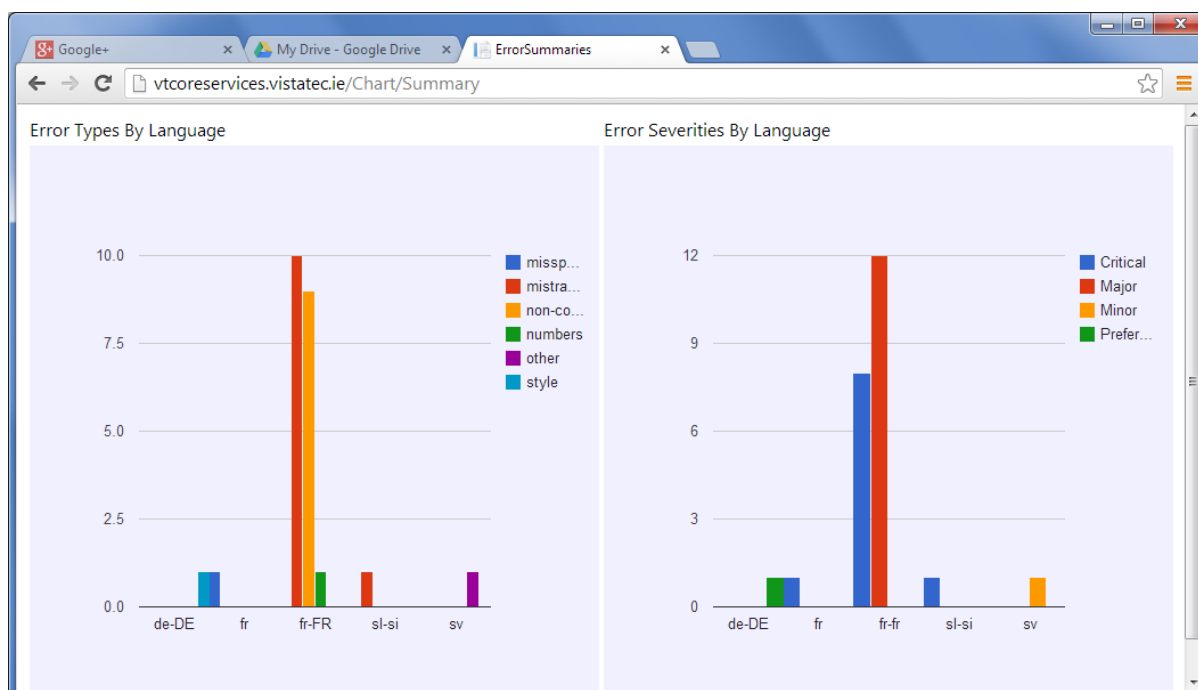


Figure 4: Quality Web Dashboard. LQI metadata can be sent via RESTful API to VistaTEC endpoint and quality metrics immediately updated in portal dashboard.

5. OCELOT OPEN SOURCE AND TECHNICAL DETAILS

VistaTEC's announced the availability of Ocelot as Open Source under the LGPL License at Localization World, Santa Clara in October 2013.

Source code is available at: <https://github.com/ocelot>

Documentation wiki is at: <http://open.vistatec.com/ocelot>

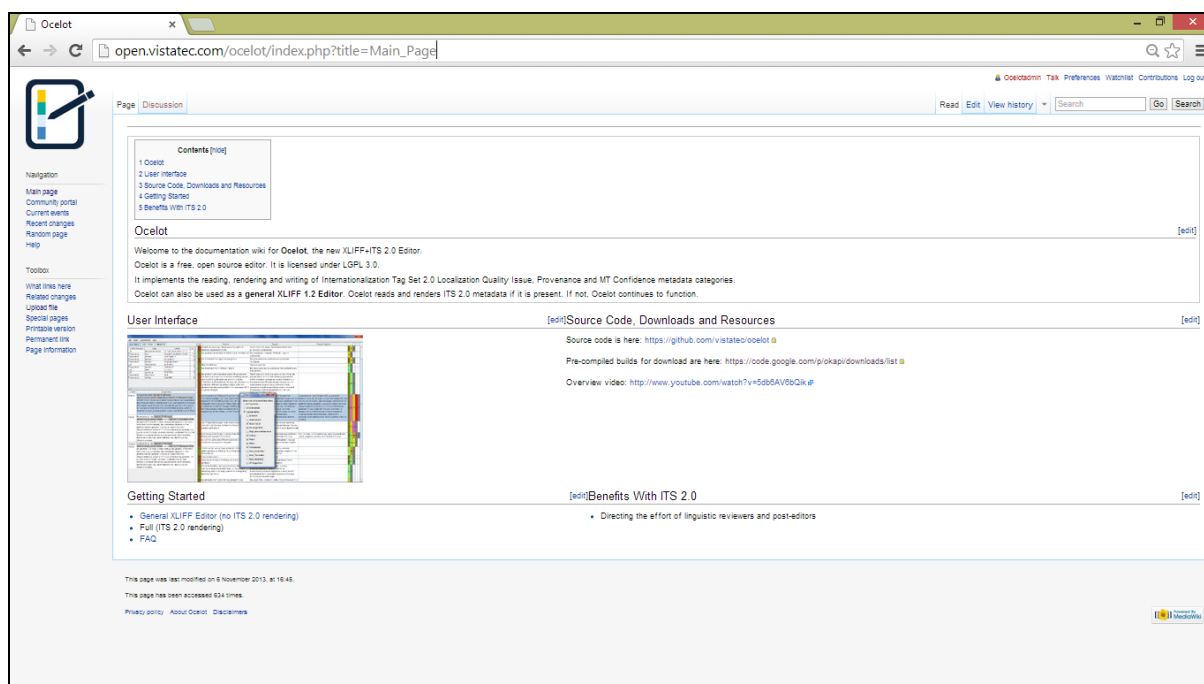
Binaries are available at:

Email list: okapitools@yahoogroups.com

Ocelot is written in Java.

It uses (and enhanced) several Okapi Framework classes, namely: `XLIFFFilter`, `XLIFFSkeletonWriter`, `ITSStandoffManager` and `ITSProvenance*`.

The easiest way of getting setup with the source code is to install the Oracle Netbeans IDE. Download the code from GitHub and open the Maven `pom.xml`. This will initiate Netbeans downloading all of the dependency libraries. Once this completes you can build and launch the application.



6. TANGIBLE OUTCOMES OF OCELOT

The technology startup Digital Linguistics (<http://www.digitallinguistics.com>) has adopted ITS 2.0 as its mechanism for writing its text analytics-generated Conformance Scores into XLIFF.

Ocelot and ITS 2.0 were presented at the TAUS Translation Quality Evaluation Summit held in Adobe on 9th October 2013. This event was attended by many of the world's top localizers.

Ocelot now forms the basis of a post-editing service within VistaTEC which is based fundamentally upon ITS 2.0 metadata.

VistaTEC's process now consists of the following steps:

1. Source files are converted to XLIFF 1.2. This can be done with Okapi Rainbow.
2. Files are sent to machine translation engine.
3. Files containing raw machine translation output are run through Okapi Checkmate. This adds ITS metadata for common errors.
4. Files are posted to Digital Linguistics' Review Sentinel service which adds "non-conformance" ITS markup.
5. Ocelot and a `rules.properties` file is distributed to post-editors. The `rules.properties` file contains rendering and segment filtering rules for 4x 10-point quality non-conformance bands. It also configures keyboard shortcut keys for adding Fluency/Adequacy scores.
6. Files are post-edited. Post-editors carry out edits using the 4 number 10-point non-conformance filters as their guide for segments to post-edit. Editors also add Fluency/Adequacy scores according to

proprietary guidelines.

7. Files are post-processed to harvest metadata added during post-editing.
8. Files are converted back to native format.