D4.3: XLIFF ROUNDTRIPPING PROTOTYPE BASED ON M4LOC WORK AND OKAPI TOOLS

Milan Karásek

Distribution: Public

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	4.3
Deliverable title:	XLIFF Roundtripping Prototype based on M4Loc Work and Okapi Tools
Dissemination level:	PU
Contractual date of delivery:	30 September 2013
Actual date of delivery:	30 September 2013
Author(s):	Milan Karásek
Participants:	Moravia, UL
Internal Reviewer:	Linguaserve
Workpackage:	WP4
Task Responsible:	UL
Workpackage Leader:	Linguaserve

Revision History

Revision	**Date**	**Author**	**Organization**	**Description**
1	30/09/2013	Milan Karásek	Moravia	Draft
2	24/10/2013	David Filip	UL	Revised Version
3	31/10/2013	Milan Karásek	Moravia	Implemented feedback from Stephan Walter (Cocomore)

Contents

Document Information 2

Revision History 2

Contents 3

1. Executive Summary 4

2. Reference Implementation 5

2.1. Base Technologies 5

XLIFF 5

Moses MT 5

M4Loc 5

Okapi Framework 6

2.2. ITS 2.0 Support 6

Domain 6

Translate 7

Text Analysis 8

MT Confidence 9

Provenance 9

2.3. Description of the Web Service 10

Processing XLIFF File (`mlwlt_xliff_mt_prepare`) 10

The list of job files (`mlwlt_job_list`) 11

The Job log (`mlwlt_job_log`) 11

XLIFF output file for previous jobs (`mlwlt_job_output`) 12

General service information (`mlwlt_web_service_information`) 12

The Echo function (`mlwlt_xliff_mt_echo`) 13

2.4. Web service implementation 13

2.5. Test Site 13

References 14

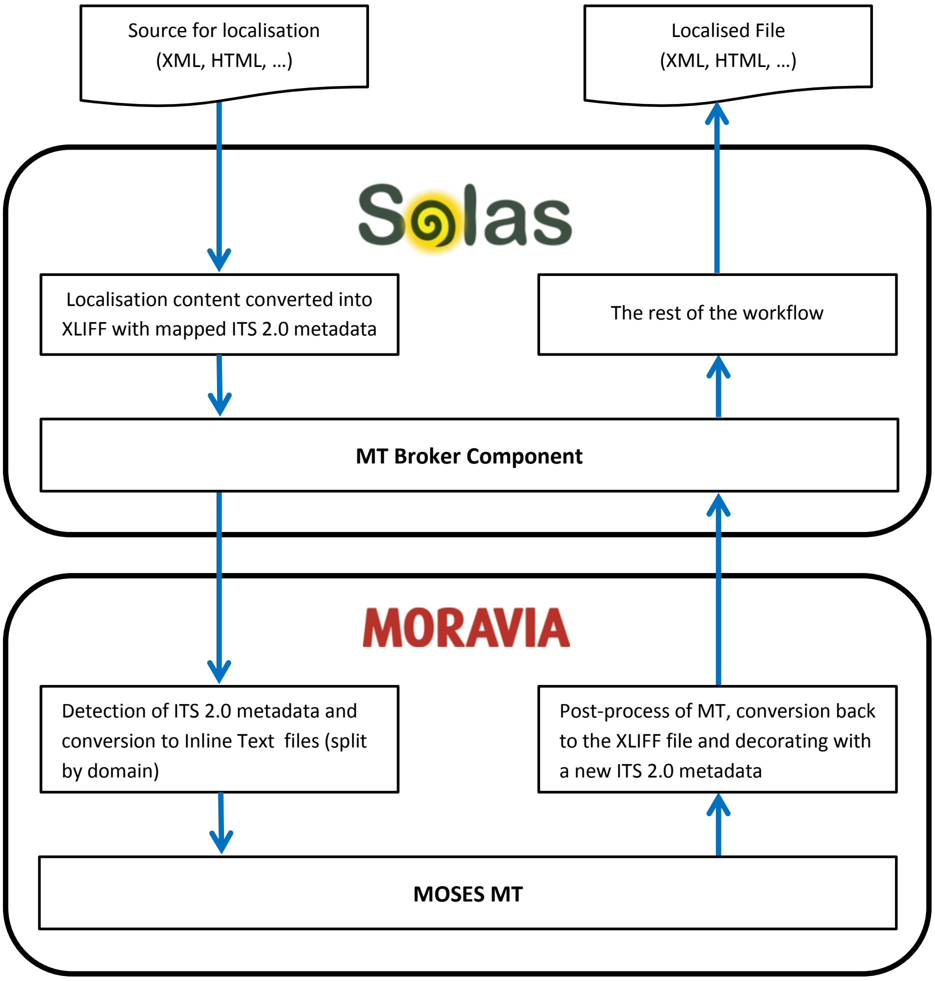
1. Executive Summary

The goal this deliverable is to implement an XLIFF roundtripping prototype, where content enhanced by ITS 2.0 metadata is converted to the XML Localisation Interchange File Format (XLIFF), machine translated and converted back into its original format. In the roundtripping process, all original ITS 2.0 metadata are preserved and even new metadata (like MT confidence) can be introduced during the machine translation processing phase.

Moravia has implemented a web service processing XLIFF files decorated by mapped ITS2.0 metadata. Based on the metadata, the implemented process decides which parts of XLIFF are going to be machine translated and eventually which specifically trained Machine Translation Engine is going to be used for given domains.

The web service consumes an XLIFF file (which is the only parameter at the input), finds supported ITS 2.0 categories within the file, prepares the localisable content for Moses MT using M4Loc tools, sends that content to the Moses MT engine and when translated, inserts machine translated text back to the XLIFF file, as a translation suggestion: using the <alt-trans> element in XLIFF 1.2 or the Translation Candidates module in XLIFF 2.0.

The web service can be called from an MT broker component as part of a SOLAS-based localisation workflow in deliverable D3.1.2 - XLIFF Roundtripping plus XSLT for Hidden Web Formats.


Source for localisation
(XML, HTML, …)
Localised File
(XML, HTML, …)
Solas
Localisation content converted into XLIFF with mapped ITS 2.0 metadata
The rest of the workflow
MT Broker Component
MORAVIA
Detection of ITS 2.0 metadata and conversion to Inline Text files (split by domain)
Post-process of MT, conversion back to the XLIFF file and decorating with a new ITS 2.0 metadata
MOSES MT

2. Reference Implementation

The reference implementation described in this deliverable has been designed as a web service which creates a job accepting XLIFF file at the input, preparing the file for M4Loc processing and translation with moses MT, and returns the XLIFF file including machine translation results, based on ITS 2.0 metadata.

2.1. Base Technologies

The following list gives some background information on each of the base technologies used in the process that has been implemented within the scope of this deliverable:

XLIFF

XLIFF (XML Localisation Interchange File Format) is an XML-based format created to standardize the way localisable data are passed between tools during a localisation process. [1][2]

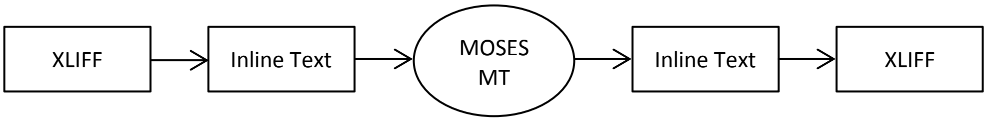
Currently, the latest official version (OASIS standard) of XLIFF is version 1.2, yet the implemented roundtripping prototype also supports the new version 2.0, which is currently (fall 2013) under a public review and is expected to become an official OASIS standard in February 2014.

MOSES MT

Moses is a statistical machine translation (SMT) system that provides automatic training capabilities for creation of custom translation and target language models for any language pair. A collection of previously translated texts (a bi-lingual parallel corpus) is used for training of the translation model. Once the trained model is created, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices. [3][4]

M4LOC

The goal of the M4Loc project is to provide tools to translate localisation-specific formats with Moses and to integrate Moses in localisation workflows. The M4Loc is dealing with the main localisation specific problem during MT processing in Moses: the in-line tags. As Moses MT can work only with plain text as input, it is normally not possible to preserve inline tags in localisation specific formats. Even if there is a way to preserve inline tags, placing them properly in the target language poses an additional challenge. The M4Loc solution ensures that localisation formats (generally inline markup rich formats such as XLIFF) can be converted into the so called “InlineText” file format which is understandable to Moses MT and (after translation) can be converted back into the originating localisation format. [5][6]



A number of tools is included in the solution including Okapi Framework Tikal.

OKAPI FRAMEWORK

The Okapi Framework is a set of interface specifications, format definitions, components and applications that provide an environment to build interoperable tools for the different steps of the translation and localisation process.

The goal of the Okapi Framework is to allow tools developers and localizers to build new localisation processes or enhance existing ones to best meet their needs, while preserving a level of compatibility and interoperability. It also provides them with a way to share (and re-use) components across different solutions. The project uses and promotes open standards, where they exist. For the aspects where open standards are not defined yet, the framework offers its own. The ultimate goal is to adopt the industry standards when they are defined and useable. [7]

2.2. ITS 2.0 Support

The web service supports a subset of ITS 2.0 categories that are important for MT processes in general and using Moses MT in particular. Some of the supported categories are accepted and processed at the input of the service, used to decide what is to be translated and how, other supported data categories are generated during the translation.

An informative recommendation on how the ITS 2.0 data categories are represented in XLIFF is being finalized by the International Tag Set Interest Group and is publicly available:

- XLIFF 1.2 Mapping: http://www.w3.org/International/its/wiki/XLIFF_1.2_Mapping
- XLIFF 2.0 Mapping: http://www.w3.org/International/its/wiki/XLIFF_2.0_Mapping

XLIFF TC is going to formalize support of ITS 2.0 as an official XLIFF 2.1 module within 2014, based on the mapping agreed between the W3C and OASIS working groups during the ITS 2.0 development.

Following ITS 2.0 categories are supported by the web service:

DOMAIN

This data category is used as an input value for the web-service. This defines the domain or domains (in the sense of topic or expertise area) to which the source text belongs. This category can be specified at all structural levels. So if needed different translation units or groups of units within the same XLIFF document can be handled by different specialized MT engines.

A value in the domain data category is used for decision which MT engine will be called for translation. The list of available MT engines and their mapping to domains is defined in the web-service configuration.

Example of using Domain data category in XLIFF 1.2:

```
<trans-unit id="1" itsx:domains="classical-studies">
    <source xml:lang="en-us">Classical Studies</source>
</trans-unit>
```

XLIFF 2.0:

```
<unit id="1">
    <segment id="segID_1" itsx:domains="classical-studies">
        <source xml:lang="en-us">Classical Studies</source>
    </segment>
</unit>
```

Note that `domains` attribute contains `itsx:` in its name, which is a schema prefix for the namespace http://www.w3.org/ns/its-xliff/.

TRANSLATE

Translate data category is used to identify parts of the XLIFF file which should (or shouldn't) be translated. Entire content of the XLIFF file is treated as translatable by default, therefore web-service is looking for segments marked as translate="no" and removes them from the content which is sent to MT.

In case there is a part of translatable text marked as translate="no" (mapped in XLIFF 1.2 as `<mrk mtype="protected">`), content of the span is marked in the M4Loc as not translatable and remains in original language while the rest of the text is machine translated.

Example of using Translate data category in XLIFF 1.2:

```
<trans-unit id="2" translate="yes">
    <source xml:lang="en-us">
        The first 'Classic' writer was Aulus Gellius a 2nd-century Roman writer
        who in the miscellany Noctes Atticae (19  8  15) refers to a writer as
        a <mrk mtype="protected">Classicus scriptor non proletarius</mrk>
        ('A distinguished  not a commonplace writer').
</source>
</trans-unit>
```

XLIFF 2.0:

```
<unit id="2" translate="yes">
    <segment id="seg_1">
        <source xml:lang="en-us">
            The first 'Classic' writer was Aulus Gellius a 2nd-century Roman writer
            who in the miscellany Noctes Atticae (19  8  15) refers to a writer as
            a <mrk id="mrk_1" translate="no">Classicus scriptor non proletarius</mrk>
            ('A distinguished  not a commonplace writer').
        </source>
    </segment>
</trans-unit>
```

TEXT ANALYSIS

The web service detects Text Analysis data category metadata in the translatable content. When the identifier of the text analysis target (`taIdentRef` attribute) is defined, service is looking for translation at the referred target instead of MT. Translation using text analysis reference is more accurate than the machine translation. Current implementation supports all references to http://www.dbpedia.org site as an analysis target.

In case that text analysis target is not defined or there is not possible to get translation from that target, original word is sent to the Moses MT for translation.

Example of using Text Analysis data category in XLIFF 1.2:

In the sentence ‘From the canyons of Arizona or from the wild savannah of Africa.’ have been detected two words by Text Analysis tool. Both of them has also defined `taClassRef` attribute which is used to obtain the translation.

```
<trans-unit id="3" itsx:domain="IT">
    <source xml:lang="en-us">
        From the canyons of
        <mrk mtype="x-its" its:taConfidence="0.7"
          its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
          its:taIdentRef="http://dbpedia.org/resource/Arizona">Arizona</mrk>
        or from the wild savanna of
        <mrk mtype="x-its" its:taConfidence="0.7"
          its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
          its:taIdentRef="http://dbpedia.org/resource/Africa">Africa</mrk>.
    </source>
</trans-unit>
```

XLIFF 2.0:

```
<unit id="3" itsx:domain="IT">
    <segment id="seg_1">
        <source xml:lang="en-us">
            From the canyons of
            <mrk id="mrk_1" mtype="x-its" its:taConfidence="0.7"
              its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
              its:taIdentRef="http://dbpedia.org/resource/Arizona">Arizona</mrk>
            or from the wild savanna of
            <mrk id="mrk_2" mtype="x-its" its:taConfidence="0.7"
              its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
              its:taIdentRef="http://dbpedia.org/resource/Africa">Africa</mrk>.
        </source>
    </segment>
</unit>
```

MT CONFIDENCE

First of data categories which is generated by the web-service is MT confidence which represents the self-reported confidence score from a machine translation engine of the accuracy of a translation it has provided.

Example of using MT Confidence data category in XLIFF 1.2:

The MT confidence related attributes are `match-quality` representing the confidence score and `its:annotatorsRef` to identify the origin of the MT translation.

```
<alt-trans match-quality="0.749" origin="MT" its:provenanceRecordsRef="#pr3"
  its:annotatorsRef="mtconfidence|http://mlwlt.moravia.com/mlwlt-service-xliff-mt">
    <target xml:lang="Spanish">
        La primera 'Classic' Writer era aulus gellius un 2nd-century Roman Writer
        que en la miscellany noctes atticae (19 8 15) refers a una Writer como un
        <mrk mtype="protected">Classicus scriptor non proletarius</mrk>
        ('un Distinguished no una moneda corriente Writer').
    </target>
</alt-trans>
```

PROVENANCE

Provenance data category is used to identify the source of the MT and the identification of the web-service for tracking and further processing in the localisation workflow. This data category is also added to the XLIFF by the web-service.

Example of using Provenance data category in XLIFF 1.2:

```
<xliff version="1.2">
  <file>
    <header>
      <its:provenanceRecords xml:id="pr3">
        <its:provenanceRecord its:tool="mosesmt" its:orgRef=http://www.moravia.com
          its:provRef="http://mlwlt.moravia.com/mlwlt-web-test/LogDetailXml.aspx?log=
            2013-06-06%2018.39.56.719"/>
      </its:provenanceRecords>
    </header>
    <body>
      ...
      <alt-trans match-quality="0.749" origin="MT" its:provenanceRecordsRef="#pr3" ...
      ...
    </body>
  </file>
</xliff>
```

XLIFF 2.0:

Encoding in XLIFF 2.0 is the same as in version 1.2. Only difference is in placing its:provenanceRecords directly under the <file> element as there is no <header> section in XLIFF 2.0.

2.3. Description of the Web Service

The web-service has implemented following functions:

mlwlt_xliff_mt_prepare	The main function for processing of the XLIFF file.
mlwlt_job_list	Returns a list of previously created jobs.
mlwlt_job_log	Returns a log for given Job ID.
mlwlt_job_output	Returns an XLIFF output for any previously created job.
mlwlt_web_service_information	Returns general information about the web service.
mlwlt_xliff_mt_echo	For testing purposes. Function simply returns XLIFF file from the input with no change on it.

PROCESSING XLIFF FILE (MLWLT_XLIFF_MT_PREPARE)

Function mlwlt_xliff_mt_prepare is a main function of the web service, calling from the outside to process XLIFF file. The function accepts two parameters at the input: Name of the XLIFF file and its binary data. Once the file is uploaded to the web server, function detects the version of the XLIFF and starts a new job for file translation.

Each translation is ITS 2.0 driven. It means that mapped ITS 2.0 metadata are used for localisable content detection, domain recognition and text analysis processing. As implemented, the file goes through following process:

1. Detection of domains takes place first. If XLIFF file contains mapped information about ITS 2.0 domain category (itsx:domains attribute), XLIFF file is split to several files each for given domain.

2. Each "domain file" is then prepared for the machine translation using M4Loc process. It represents conversion from the XLIFF format into the Inline Text file format, suitable for Moses MT.

   Conversion of the XLIFF version 1.2 is done using Okapi Tikal tool, but the XLIFF 2.0 is converted with the web service itself (while this version of XLIFF is not supported by Okapi yet).

   At this stage, Inline Text format still contains an information about Translate and Text Analysis data categories, mapped to the in-line spans of text.

3. Detection of the Translate data category in the Inline Text and marking the text as non-translatable, using <n> tag: By default, Moses is translating entire text from the input, except the <n> tag which is

processed differently. When a part of translatable text is encapsulated by the <n> tag, Moses MT is looking into it's translate attribute and uses the content of this attribute instead of machine translation of encapsulated text. This behaviour is used while processing both Translate and Text Analysis data categories.

When the in-line span of text is marked with translate="no", its original text is inserted to the translate attribute.

4. Detection of the Text Analysis data category in the Inline Text and looking at referenced site (when available) for proper translation. If the translation is obtained from referenced data source, it is inserted directly in the translate attribute of the <n> tag and thus propagated to the translated output.

5. Prepared Inline Text file is sent to the Moses MT engine for translation. For reference implementation purposes there are trained two language pairs (EN>FR and EN>ES). For both language pairs we have implemented two engines trained on data from different domains (4 engines in general).

6. When the file is machine translated, Inline Text format is converted back into XLIFF and all domain specific XLIFF files are merged back to the original XLIFF file.

7. XLIFF is decorated by MT Confidence and Provenance metadata and sent to the output.

THE LIST OF JOB FILES (MLWLT_JOB_LIST)

The web service stores all translation jobs at the server. Using function MLWLT_JOB_LIST, we can get the list of all stored Job IDs. The Job ID is a unique identification of each job, representing an exact time of job creation. Function MLWLT_JOB_LIST returns an XML formatted list of Job IDs, like in this example:

```
<jobs>
    <job>2012-10-18 11.50.54.486</job>
    <job>2013-09-13 16.54.15.643</job>
    <job>2013-09-13 12.59.39.642</job>
    <job>2013-09-07 07.05.50.366</job>
</jobs>
```

Function MLWLT_JOB_LIST has no input parameters.

THE JOB LOG (MLWLT_JOB_LOG)

Each job has attached its log file, describing all actions that took place during the job processing. Function MLWLT_JOB_LOG purpose is to access particular job log. The only input parameter is the Job ID; output is an XML formatted log of actions. Each action entry in the log has its time-stamp (date attribute), name of the ITS domain to which is the entry related and the phase name (phase attribute).

Example of the Job Log:

```
<log>
    <entry date="2013-10-22T10:58:39.565" domain="All" phase="Start">Log created.</entry>
    <entry date="2013-10-22T10:58:39.570" domain="All" phase="Input File Upload">Input file uploaded (demo.xlf)</entry>
    <entry date="2013-10-22T10:58:39.573" domain="All" phase="Input File Format">Input file format detected (XLF)</entry>
    <entry date="2013-10-22T10:58:39.590" domain="All" phase="Input File Format">Input XLIFF file checked: XLIFF 2.0 Detected</entry>
    <entry date="2013-10-22T10:58:39.591" domain="All" phase="Input File Format">Input XLIFF file format: 2.0</entry>
    <entry date="2013-10-22T10:58:39.593" domain="All" phase="Provenance">Created a new provenance id: 1</entry>
    <entry date="2013-10-22T10:58:39.603" domain="Legal" phase="Domain Mapping Detection">Detected domain: DEFAULT</entry>
    <entry date="2013-10-22T10:58:39.604" domain="IT" phase="Domain Mapping Detection">Detected domain for ITS domain name: IT</entry>
    <entry date="2013-10-22T10:58:39.609" domain="Legal" phase="XLIFF Split">XLIFF has been split for given domain.</entry>
    <entry date="2013-10-22T10:58:39.612" domain="IT" phase="XLIFF Split">XLIFF has been split for given domain.</entry>
    <entry date="2013-10-22T10:58:39.618" domain="Legal" phase="M4Loc">XLIFF 2.0 converted to the in-line text format</entry>
    <entry date="2013-10-22T10:58:39.620" domain="Legal" phase="ITS Translate">Protected in-line text: "this couple of words"</entry>
    <entry date="2013-10-22T10:58:41.651" domain="Legal" phase="ITS TextAnalysis">TextAnalysis detected ("Arizona")</entry>
    <entry date="2013-10-22T10:58:42.432" domain="Legal" phase="ITS TextAnalysis">TextAnalysis detected ("Antarctica")</entry>
    <entry date="2013-10-22T10:58:42.891" domain="Legal" phase="ITS TextAnalysis">TextAnalysis detected ("Africa")</entry>
    <entry date="2013-10-22T10:58:42.895" domain="Legal" phase="M4Loc">The in-line text file has been sent to Moses MT engine</entry>
    <entry date="2013-10-22T10:59:30.676" domain="Legal" phase="M4Loc">The in-line text file has been translated by Moses MT</entry>
    <entry date="2013-10-22T10:59:30.685" domain="IT" phase="M4Loc">XLIFF 2.0 converted to the in-line text format</entry>
    <entry date="2013-10-22T10:59:30.688" domain="IT" phase="M4Loc">The in-line text file has been sent to Moses MT engine</entry>
    <entry date="2013-10-22T10:59:37.626" domain="IT" phase="M4Loc">The in-line text file has been translated by Moses MT</entry>
    <entry date="2013-10-22T10:59:37.633" domain="Legal" phase="XLIFF Merge">Domain XLIFF has been merged.</entry>
    <entry date="2013-10-22T10:59:37.635" domain="IT" phase="XLIFF Merge">Domain XLIFF has been merged.</entry>
    <entry date="2013-10-22T10:59:37.636" domain="All" phase="Output File">Output file is successfully returned</entry>
</log>
```

XLIFF OUTPUT FILE FOR PREVIOUS JOBS (`MLWLT_JOB_OUTPUT`)

Function to get translated XLIFF file from a job that was previously translated. The input parameter is the Job ID and the output is the XLIFF file.

GENERAL SERVICE INFORMATION (`MLWLT_WEB_SERVICE_INFORMATION`)

The web service is using a configuration file containing information needed for a proper functionality of the service as well as the information for identification of the service, like the supported ITS 2.0 data categories, supported language pairs and domain mapping. Based on the configuration, foreign systems can decide whether the web service is suitable for them or not.

Function returns the configuration in XML format.

In the example (see on the right side) you can see the section of the configuration, related to the supported ITS 2.0 data categories (`<supported-its-categories>` element) with the list of named categories.

Section Domain mapping (`<domain-mapping>` element) contains supported language pairs with implemented domains in them (different MT engine is trained for each domain). For example language pair English to French has implemented two domains: IT and Legal. The IT domain is used when ITS domain category uses value 'IT' or 'classical-studies'.

```
<configuration>
    <supported-its-categories>
        <category>Translate</category>
        <category>Text Analysis</category>
        <category>Domain</category>
        <category>MT Confidence</category>
        <category>Provenance</category>
    </supported-its-categories>
    <domain-mapping>
        <language-pair source-language="English"
                       target-language="French"
                       default-domain="Legal">
            <domain name="Legal">
                <its-domains>
                    <its-domain>classical-quotes</its-domain>
                    <its-domain>travel</its-domain>
                </its-domains>
            </domain>
            <domain name="IT">
                <its-domains>
                    <its-domain>IT</its-domain>
                    <its-domain>classical-studies</its-domain>
                </its-domains>
            </domain>
        </language-pair>
        <language-pair source-language="English"
                       target-language="Spanish"
                       default-domain="Legal">
            <domain name="Legal">
                <its-domains>
                    <its-domain>classical-quotes</its-domain>
                    <its-domain>travel</its-domain>
                </its-domains>
            </domain>
            <domain name="IT">
                <its-domains>
                    <its-domain>IT</its-domain>
                    <its-domain>classical-studies</its-domain>
                </its-domains>
            </domain>
        </language-pair>
    </domain-mapping>
</configuration>
```

THE ECHO FUNCTION (MLWLT_XLIFF_MT_ECHO)

The echo function has been implemented for web service integration testing purposes. The function accepts the same parameters as the main function for XLIFF processing (Name of the XLIFF file and its binary data) and returns XLIFF without any change.

2.4. Web service implementation

The web service is implemented and hosted at Moravia web site. The URL of the service is following:

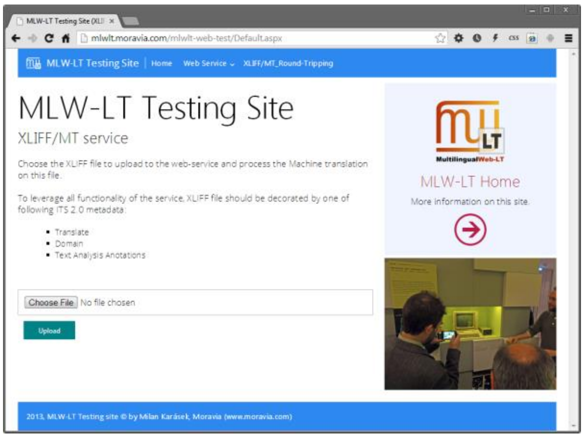
http://mlwlt.moravia.com/mlwlt-service-xliff-mt/mlwlt-service.asmx

The service description (WSDL) is available here:

http://mlwlt.moravia.com/mlwlt-service-xliff-mt/mlwlt-service.asmx?WSDL

2.5. Test Site

We have also available the testing site with ability to create a new job, uploading an XLIFF file to the web-based application (http://mlwlt.moravia.com/mlwlt-web-test). When the file is uploaded, new created job is immediately executed and translated XLIFF file is returned back. In the application, the complete job log is also available.



MLW-LT Testing Site

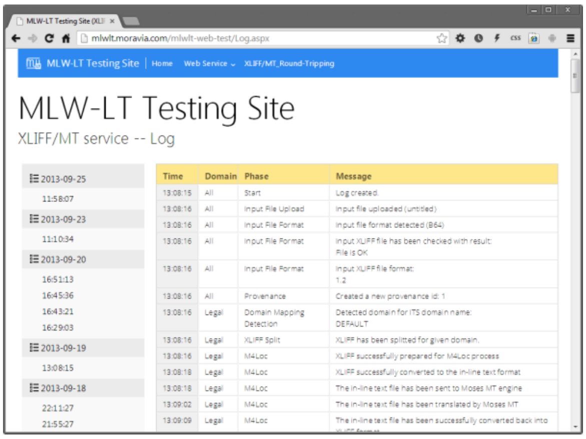
XLIFF/MT service

Choose the XLIFF file to upload to the web-service and process the Machine translation on this file.

To leverage all functionality of the service, XLIFF file should be decorated by one of following ITS 2.0 metadata:

- Translate
- Domain
- Text Analysis Anotations

2013, MLW-LT Testing site © by Milan Karásek, Moravia (www.moravia.com)



MLW-LT Testing Site

XLIFF/MT service -- Log

Time	Domain	Phase	Message
13:08:15	All	Start	Log created.
13:08:16	All	Input File Upload	Input file uploaded (untitled)
13:08:16	All	Input File Format	Input file format detected (B64)
13:08:16	All	Input File Format	Input XLIFF file has been checked with result: File is OK
13:08:16	All	Input File Format	Input XLIFF file format: 1.2
13:08:16	All	Provenance	Created a new provenance id: 1
13:08:16	Legal	Domain Mapping Detection	Detected domain for ITS domain name: DEFAULT
13:08:16	Legal	XLIFF Split	XLIFF has been splitted for given domain.
13:08:16	Legal	M4Loc	XLIFF successfully prepared for M4Loc process
13:08:18	Legal	M4Loc	XLIFF successfully converted to the in-line text format
13:08:18	Legal	M4Loc	The in-line text file has been sent to Moses MT engine
13:09:02	Legal	M4Loc	The in-line text file has been translated by Moses MT
13:09:09	Legal	M4Loc	The in-line text file has been successfully converted back into

REFERENCES

1. XLIFF definition at Wikipedia,
   http://en.wikipedia.org/wiki/XLIFF

2. OASIS XML Localisation Interchange File Format (XLIFF) TC homepage
   https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

3. Moses MT at Wikipedia
   http://en.wikipedia.org/wiki/Moses_(machine_translation)

4. Moses MT project homepage
   http://www.statmt.org/moses/

5. M4Loc project homepage
   https://code.google.com/p/m4loc/

6. Tomáš Hudík, Achim Ruopp: The Integration of Moses into Localisation Industry, EAMT 2011
   http://www.mt-archive.info/EAMT-2011-Hudik.pdf

7. Okapi Framework
   http://okapi.sourceforge.net/index.html

8. Service-Oriented Localisation Architecture Solution (SOLAS)
   http://www.localisation.ie/solas/