D4.1.4: REPORT ON MODIFICATIONS IN MT SYSTEMS

Pablo Nieto Caride, Thomas Ruedesheim, Ankit K. Srivastava, Giuseppe Deriard Nolasco, Declan Groves, Pedro L. Díez Orzas

Distribution: Public

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	4.1.4
Deliverable title:	Report on Modifications in MT Systems
Dissemination level:	PU
Contractual date of delivery:	31st January 2013
Actual date of delivery:	15st February 2013
Author(s):	Pablo Nieto Caride Nolasco, Thomas Ruedesheim, Ankit K. Srivastava, Giuseppe Deriard, Declan Groves, Pedro L. Díez Orzas
Participants:	Linguaserve, DCU, Lucy Software
Internal Reviewer:	Linguaserve
Workpackage:	WP4
Task Responsible:	Giuseppe Deriard
Workpackage Leader:	Pedro L. Díez Orzas

Revision History

Revision	Date	Author	Organization	Description
1	01/01/2012	Pablo Nieto Caride, Thomas Ruedesheim, Ankit K. Srivastava	Linguaserve, Lucysoftware, DCU	Compiling information and draft creation
2	13/02/2013	Pablo Nieto Caride, Thomas Ruedesheim, Ankit K. Srivastava	Linguaserve, Lucysoftware, DCU	Final Version
3	15/02/2013	Giuseppe Deriard, Pedro L. Díez Orzas, Declan Groves	Linguaserve	Revised Final Version

CONTENTS

1. EXECUTIVE SUMMARY

The present document constitutes a detailed report of the different modifications needed to adapt Linguaserve's Real Time Multilingual Publication System ATLAS PW1, DCU's Statistical MT System MaTrEx, and LucySoftware's Rule-based MT System to be ITS 2.0 compliant.

The first two sections consist of a brief introduction of how the different solutions provided by said real time translation systems can be immensely helpful to deal with the web multilingualism, and the current progress of the work-plan represented with a timeline, respectively.

After that, the subsequent sections illustrate a description of the real time translation systems and the modifications necessary to support ITS 2.0, along with an argumentation on how each system is expected to contribute to encourage metadata users and consumers to a widespread use of the standard.

Finally a quick guideline to become acquaintance with the utilised metadata and a list of references to be consulted in case of necessity of more in-depth information, are given in the last two sections.

2. INTRODUCTION

The large volume of web content, the speed of continuous updates and the webs 2.0 and 3.0 require real time translation systems that provide sufficient quality and precision. Metadata for linguistic technology is crucial to identify and process different linguistic elements and features in HTML web pages. A set of the metadata defined in the ITS 2.0 standard has been implemented and is used by the Online MT System to take advantage of its properties and features in order to improve the process of real time translations.

The showcase will exemplify how ITS 2.0 allows an HTML5 Content Author to send instructions on the translation to MT Systems and to a Content Editor through a Real Time Translation System (RTTS). This RTTS is connected to different MT Service Providers; more specifically, a Statistical MT System MaTrEx (DCU) and a Rule-based MT System (LucySoftware).

3. OBJECTIVES AND WORK PLAN

The objective of the report is to describe the different module modifications and developments corresponding to the deliverables 4.1.1, 4.1.2 and 4.1.3., Lucy Modification, MaTrEx Modification, and Linguaserve Online System Modification respectively.

The work plan was organised taking into account the following criteria:

1. Active participation in the ITS 2.0 requirements and definition phases.
2. Delaying of the metadata implementation development as much as possible (last December specification version) to include the latest metadata features.
3. Contribution in the creation and correction of several Test Suite's metadata input files as well as the implementation of various data category features with their respective output files.
4. Coordinating with subcontracted third party development (Daedalus)

A three level approach guided the work:

- ITS 2.0 definition level
    - Applicability of ITS 2.0 (data category selection)
- System level
    - Proper requirements and needs for each system: independent system ITS 2.0 compliance.
- Solution level
    - Connectivity between online multilingual publication system and the machine translation system
    - ITS 2.0 Interoperability between online multilingual publication system and the machine translation system: global solution ITS 2.0 compliance
    - Getting ready to develop the showcase in 2013 (Deliverable 4.2.1 and 4.2.2)

Finally, EDITA subproject was carried out in parallel and has already been completed. The results are included in this deliverable as the following two annexes:

- Annex I: MT post-editing methodology
- Annex II: MT post-editing training for translators

The next step is the development of the Online MT System Linguaserve Showcase which will be delivered in two phases:

INITIAL VERSION

The final client will begin to introduce the ITS 2.0 metadata on their own system with Linguaserve's assistance and consultancy. The final language pairs will be ES>EN in the deployment environment (final approval of the client required) and ES>FR and ES>DE in the preproduction environment.

The client is the Spanish Tax Agency (http://www.agenciatributaria.es). The selected parts and contents of the www.aeat.es web site will be annotated with ITS 2.0 to be used by the online MT system.

The following post-editing tasks applying EDITA's methodology and training will be completed:

- A full English version with Lucy MT.
- A full data category coverage sampling of the English version with MaTrEx, and the French and

German versions with Lucy MT.

FINAL VERSION

This version will be available by mid-2013, along with the Deliverable 4.2.2. Report on Online MT System.

Throughout this period of time, the selected content updates of the http://www.agenciatributaria.es website will be managed by the Online MT System and the following versions will be delivered:

- An initial English version of http://www.agenciatributaria.es will be later publically available on the Internet, using ATLAS PW1 and Lucy MT (final approval of the client required).
- An initial English version of http://www.agenciatributaria.es will be accessible in preproduction for demo purposes (with login and password), using ATLAS PW1 and MaTrEx.

A partial German and French version of http://www.agenciatributaria.es will be accessible in preproduction for demo purposes (with login and password), using ATLAS PW1 and Lucy MT.

4. Lucy Modification

4.1 Background description

The Lucy LT Engine is a rule-based MT engine bundled with a text handling module for various document formats. It also contains a basic translation memory (TM) module for pure text.

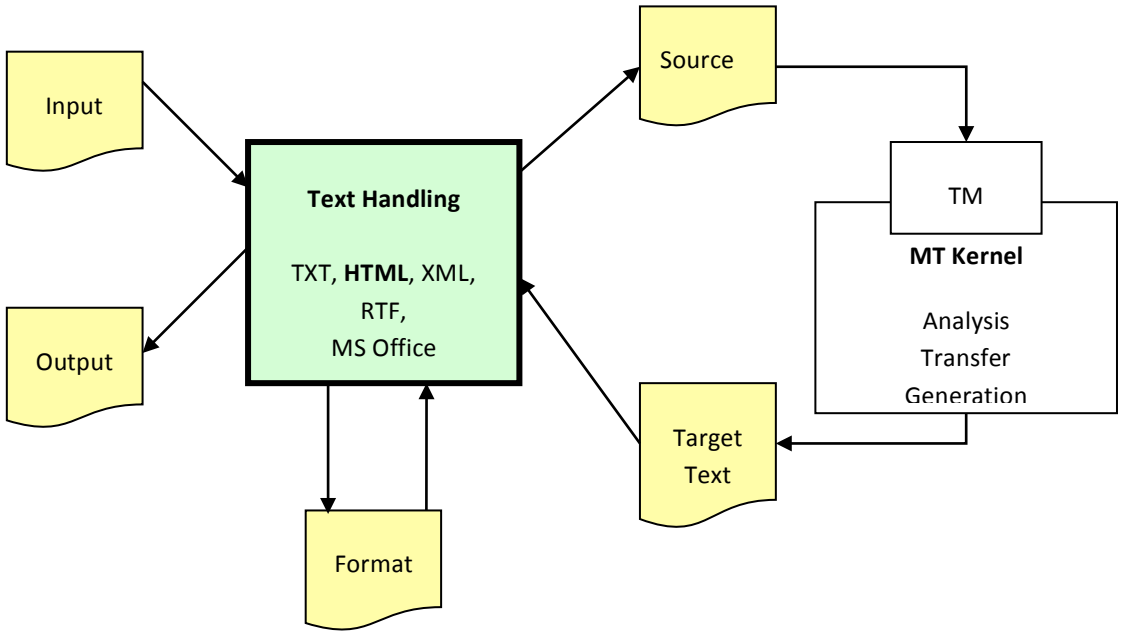

Figure 4.1: LT Engine Architecture - Workflow

The general workflow is as follows:

1. An input document is passed to the Text Handling module of LT Engine
2. A format-specific text handling routine separates textual content of the input from format information and stores them separately (Deformatting).
3. The pure textual content is passed over to the MT kernel for translation.
4. The MT kernel writes the target text into a file.
5. The text handling module merges the format information with the target text (Reformatting) and produces the output document.
6. The output document is returned to the client.

A set of translation parameters controls the whole processing of documents. The unit of processing is a document, which means that parameters have a document-wide scope.

De- and Reformatting of HTML documents are implemented as a proprietary pseudo-parser that is capable of handling a configurable set of HTML 4 tags. Translation of attributes is possible for some hard-coded attribute

names (title, summary, alt).

4.2 Foreground development description

While the overall translation process remains unchanged, the HTML part of the Text Handling module gets re-implemented using the free XML and HTML parser “libxml2”, a library that has been developed for the Gnome project, and the internal DOM structure that offers makes it easy to apply XPath expressions to the document and to manipulate document nodes.

Since it is quite challenging to replace a working (sub-) module in a productive system, the plan is a step-wise implementation:

1. Self-restraint to implement ITS 2.0 only for HTML and only for metadata that is obviously relevant for rule-based machine translation: Translate and Domain, and additionally, support of the native HTML 5 “lang” attribute.

2. In a second step, the implementation of ITS 2.0 for XHTML and XML, supporting the “xml:lang” attribute and extension of the set of supported ITS 2.0 metadata, will be performed.

4.2.1 REQUIREMENTS AND SPECIFICATIONS

All the information, specifications and requirements about the metadata can be found on the W3C ITS 2.0 Draft: http://www.w3.org/TR/2012/WD-its20-20121206/

The metadata to be implemented in the Working Package are the following:

TRANSLATE

The Translate data category indicates whether the content of an element or attribute should be translated or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

More info on: http://www.w3.org/TR/2012/WD-its20-20121206/#trans-datacat

DOMAIN

The Domain data category is used to identify the domain of the content.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#domain

4.2.2 ARCHITECTURE

The ITS 2.0 data categories are implemented in the HTML converter routines (de-formatting and re-formatting) of the Text Handling module (see Figure 1: LT Engine Architecture).

The de-formatter handles:

- Segmentation: Splitting the input into translation units (sentences).
- For HTML: Mark tags as data and separate them from text content

- Handling of abbreviations, chapters and chapter numberings
- Normalization of spaces
- Separation of font markers
- Constant marking via regular expressions and Perl filters
- Handling of translated/un-translated proper names
- TM search for already translated segments (optional)

These processing steps result in the input file for the translation kernel (see Figure 1). The (main) output file of the translation process is then passed to the re-formatter routines which handle:

- Merging of TM contents with translation results (optional)
- Copy font information back onto the target
- Merges other format information and handle internal markups
- Construct valid output format

4.2.3 FUNCTIONAL ANALYSIS

Data categories may be used locally (attributes) or globally in external XML rule files or inline rules (inside XML scripts). The specifications for handling the data categories (default values, overriding and inheritance) are contained in the ITS 2.0 requirements specification draft (see above 4.2.1).

The handling of the standard HTML “lang” attribute is made in the following way: Given a translation direction for the translation task, the system translates only text content that is not marked by a “lang” attribute or that is marked as being in the current source language. The “lang” attribute is inherited to sub-elements and may be overwritten. If text content is marked, the mark-up value will be changed to the current target language after translation took place.

4.3 ITS 2.0 implementation

IMPLEMENTED CATEGORIES

- Translate (local, global external rules, global inline rules) for HTML5
- Domain (global external rules, global inline rules) for HTML5
- Language Info (local) for HTML5

LIMITATIONS

Only the default query language XPath 1.0 is supported.

The current design of the Text Handling module stipulates to handle an input document as a whole. Translation parameters passed with a translation task to the engine are global. The domain is passed in such a global translation parameter. It cannot change throughout a document. So, the implementation ignores the

selector of domain rules and assumes that the whole document belongs to the given domain.

4.4 Contribution to the Showcase

The latest rule-based MT Engine (RBMT) that implements ITS 2.0 as described above will be delivered. That engine will run as a component of an LT TaskScheduler server installation allowing remote access via web services.

In addition, partners are being advised on how to use the APIs and workflows in the context of RTTS (Real Time Translation System).

The LT TaskScheduler provides a RESTful web service that offers both synchronous and asynchronous translation by our RBMT Engine. The service description can be downloaded in HTML as part of the service. The web service allows full control of the translation process via translation parameters.

Lucy has made a commitment to implement the data categories Translate and Domain, only for HTML5. The implementation for XML will follow but not as deliverable of this project.

The LT Engine may also be accessed locally via a client called “LT AnyWhere”. That client provides translation of short texts, Windows clipboard contents and documents (files). It can be used to verify the correct implementation of ITS 2.0 data categories.

5. MaTrEx Modification

5.1 Background description

The Dublin City University (DCU) Machine Translation (MT) system, also known as **MaTrEx** (Machine Translation using Examples) is a hybrid multi-engine MT system (implementing statistical (phrase-based / tree-based) and example-based paradigms). For the MLW-LT project, MaTrEx uses the phrase-based statistical MT (PB-SMT) engine based on the open source log-linear phrase-based decoder Moses (http://www.statmt.org/moses).

A **PB-SMT** system in contrast to a rule-based MT system is data-driven and is trained on a large corpus of sentence-aligned bilingual text (text written in source language (input) and their corresponding human translations in target language (output)). PB-SMT systems extract knowledge in the form of phrase pairs (sub-sentential alignments) from the sentence-aligned bilingual text (training data). A number of probabilistic attributes (features) for each of these phrase pairs are defined. Some of the features used in MaTrEx include source – target translation conditional probabilities (**translation model**), target language probabilities (**language model**), and source – target relative order conditional probabilities (**reordering model**). Such features are then combined in a log-linear model, the **coefficients** of which are optimized on an objective function measuring translation quality.

Once the MaTrEx engine has been trained, in order to translate (also known as decoding), the following workflow is followed:

1. An input text (in source language) is segmented into a number of phrases (consecutive sequence of words);
2. Each phrase is looked up in the translation model to compute its translation (in target language) and the phrases are scored using the log-linear features;
3. The target language sentence is composed and output along with an MT confidence score.

Thus, the input to the MaTrEx system consists of:

1. Input text (source language)
2. Source Language Code
3. Target Language Code
4. Translation Model, Language Model, Reordering Model, Coefficients of the Log-linear Model (Default settings / set by source and target language code).

The output of the MaTrEx system consists of:

1. Output text (target language);
2. MT confidence score denoting the probability of the particular translation.

5.2 Foreground development description

While the underlying MaTrEx system remains unchanged, a sequence of pre-processing and post-processing scripts are added to the framework to make it ITS-compatible for the MLW-LT project.

The original MaTrEx system only translated single sentences. It is now capable of translating complete webpages (HTML and XML).

The pre-processing module is written in PERL and takes as input a document or a segment and splits it into sub-segments based on the ITS metadata tags. Each individual sub-segment is then passed to the decoder for translation.

The post-processing module is also written in PERL. It merges all the translated sub-segments and composes the document with all the required ITS metadata tags to be generated.

5.2.1 REQUIREMENTS AND SPECIFICATIONS

All the information, the specification and the requirements about the metadata can be found in the W3C ITS 2.0 Draft: http://www.w3.org/TR/2012/WD-its20-20121206

The metadata to be implemented in the Working Package are the following:

TRANSLATE

The Translate data category indicates whether the content of an element or attribute should be translated or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#trans-datacat

LANGUAGE INFORMATION

The Language information data category is used to identify the language of a segment in the document.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#language-information

DOMAIN

The Domain data category is used to identify the domain of the content.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#domain

LOCALE FILTER

The Locale Filter data category is used to specify that a node is only applicable to certain locales (useful in localization).

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#LocaleFilter

MT CONFIDENCE

The MT Confidence data category is used to communicate confidence in the quality of the translation (output by the MaTrEx system).

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#mtconfidence

5.2.2 ARCHITECTURE

As referenced at the beginning of section 5.2, the architecture of the MaTrEx system consists of an input document splitter module (pre-processing), a translation module (MaTrEx), and an output document merging module (post-processing).

The pre-processing singles out the translatable content to be sent to the MaTrEx decoder, while the post-processing concatenates all sub-segments together into one document.

5.2.3 FUNCTIONAL ANALYSIS

The MaTrEx system handles both XML and HTML documents, both global and local, including embedded and linked documents. We have passed all 49 test suite commitments.

5.3 ITS 2.0 implementation

The current MaTrEx configuration can handle the following ITS 2.0 data categories:

- Translate
- Language Information
- Domain
- Locale Filter
- MT Confidence

5.4 Contribution to the Showcase

The DCU MT web service for MLW-LT has been implemented using the Soaplab web services software framework (http://soaplab.sourceforge.net/soaplab2/). It provides a RESTful web service. The MaTrEx web service can also be accessed directly by logging on to the website as shown below.

http://www.cngl.ie/mlwlt/

6. LINGUASERVE ONLINE SYSTEM MODIFICATION

6.1 Background description

Linguaserve Online System is a solution for multilingual publications in real time through the Internet. The system is fully scalable and configurable, allowing multi-server redundant configurations, and can be integrated in the client's Web site. The system allows the user to navigate the client's Web site in a completely transparent way.

The overall system is composed by four modules (applications).

ATLAS PW1

It is a Web application that manages the translation requests sent by the client's Web server. The system receives a translation request, downloads the original document, checks whether it was translated before or not so as to retrieve it from the cache system or to send it to the MT, and finally serves the translated document in the user's browser. ATLAS PW1 also has an administration interface that allows administrators or project managers to manage and configure projects, that is, specific tailored configurations for the different clients' needs.

MT SYSTEM

It is a rule-based or statistical MT System that can use plain text translation memories.

FILEMANAGEMENT TOOL (FMT)

It is an application that is responsible for managing the transfer of files from the MT System to the Content Editor and for updating the MT's translation memories and glossaries.

CONTENT EDITOR

It is an application that allows the edition of the translations performed by the MT System.

Here is a diagram showing the basic workflow of the overall system:

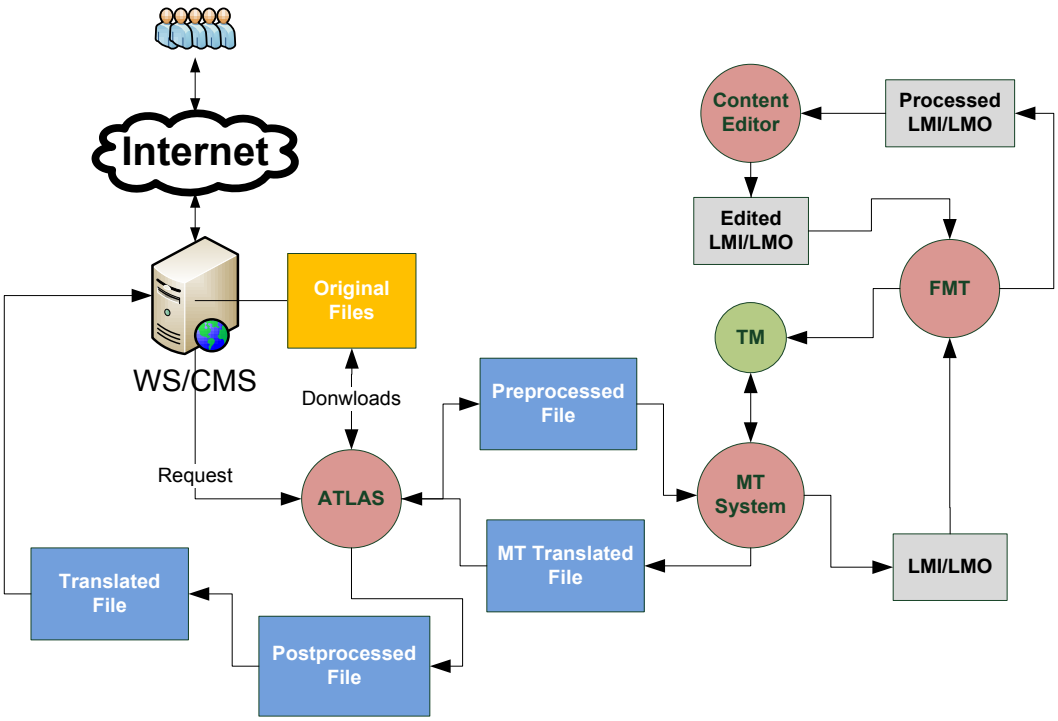

Figure 6.1: Online System Architecture - Workflow

In short, the workflow is as follows:

- The user makes a translation request and the client's Web Server redirects the request to ATLAS PW1.
- ATLAS PW1 receives a translation request, downloads the original document, and checks whether it was translated before or not so as to retrieve it from the Cache System or to send it to the MT System.
- ATLAS PW1 also pre-processes the file if the client's project configuration says so.
- If the content of the original document, or piece of the document, remains unchanged since the last translation request processed for that very document, the system retrieves the already translated content from the system's database.
- If the content of the original document is new or was not been previously processed then is sent to MT System.
- The MT System, after parsing the content to generate the list of translatable segments, checks if any of those segments is already stored in the TM (Translation Memory) to retrieve them and subsequently translates the remaining segments; here starts a sub-process related to the post-edition.
    - For all of those who are not stored in the TM, the MT System creates a pair of input and output files (LMI and LMO), which are processed and conveyed to the Content Editor by the FMT.

- Once in the Content Editor after the content of these input and output files is edited and validated, new LMI and LMO files are created and used by the FMT to update the TM.

- The MT System sends the translated content back to ATLAS PW1.
- ATLAS PW1 also post-processes the file if the client's project configuration says so.
- Finally the translated document is shown in the user's browser.

6.2 Foreground development description

While the overall process of serving translation requests to the user remains the same, with the modifications introduced in the system to be ITS 2.0 compliant, it will allow to:

- Translate HTML5 documents from an ITS-conformant Web CMS.
- Control precisely, which sentences or sentence fragments should or should not be translated and which is the source language.
- Choose, at a paragraph, sentence or word level, the appropriate training corpora or glossary (depending on the MT System) that should be used on the translation by the MT Systems in order to disambiguate.
- Convey information about the localization process to editors.
- Communicate the identity of agents that have been involved in the revision and the translation of the content, and to allow translation quality reviewers, to evaluate how the performance of these agents affects the quality of the translation.

REQUIREMENTS AND SPECIFICATIONS

All the information the specification and the requirements about the metadata can be found on the W3C ITS 2.0 Draft: http://www.w3.org/TR/2012/WD-its20-20121206

The next data categories are out of scope and they are not going to be implemented:

- Terminology
- Directionality
- Ruby
- Elements Within Text
- Disambiguation
- Locale filter
- External Resource
- Id Value

- Preserve Space
- Localization Quality Rating
- MT confidence
- Allowed Characters
- Storage Size

The metadata to be implemented in the Working Package are the following:

TRANSLATE

The Translate data category indicates whether the content of an element or attribute should be translated or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#trans-datacat

LOCALIZATION NOTE

The Localization Note data category is used to communicate notes to localizers about a particular item of content.

Two types of informative notes are needed:

- An alert contains information that the translator must read before translating a piece of text. Example: an instruction to the translator to leave parts of the text in the source language.
- A description provides useful background information that the translator will refer to only if they wish. Example: a clarification of ambiguity in the source text.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#locNote-datacat

LANGUAGE INFORMATION

The Language Information data category informs of the language of the different elements of the document.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#language-information

DOMAIN

The Domain data category is used to identify the domain of the content.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#domain

PROVENANCE

The Provenance data category is used to communicate the identity of agents that have been involved in the translation of the content or the revision of the translated content.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#provenance

LOCALIZATION QUALITY ISSUE

The Localization Quality Issue data category is used to express information related to localization quality assessment tasks.

More info on: http://www.w3.org/TR/2012/WD-its20-20121206#lqissue

ARCHITECTURE

With respect to the architecture of the system, three basic architecture diagrams are detailed in this section.

In the first place, a high level diagram of the system's architecture with a basic hardware configuration.

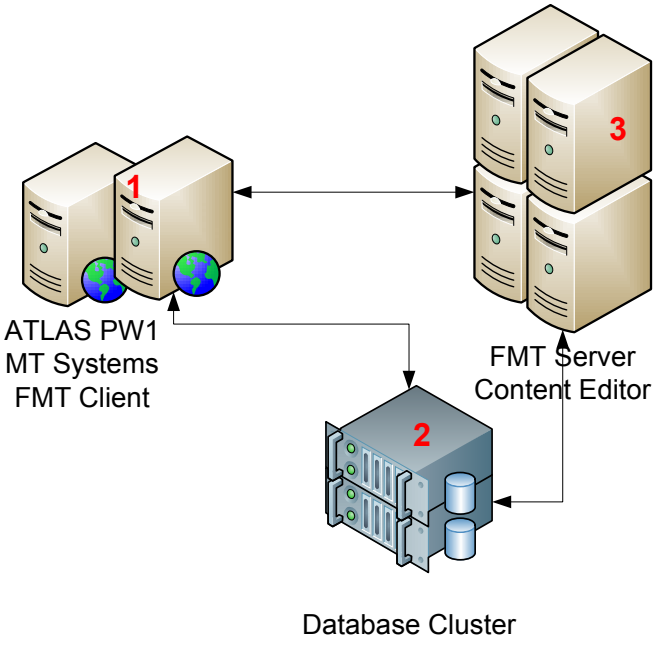

Figure 6.2: Basic hardware configuration

As shown in the above figure, the Online System can be set up in multi-server cluster configuration allowing high availability and redundancy. This is the recommended configuration but the Online System's dynamicity allows installing and running all the modules on a single server. In this figure the main modules are divided in three kinds of servers; firstly (1) Web Servers with an ATLAS PW1, an MT System and a FMT thin client installation per server, secondly (2) a Database on Cluster, and thirdly (3) an Application Server where the Content Editor and the FMT Server reside.

The next figure shows a logical point of view of the application with a module structure.

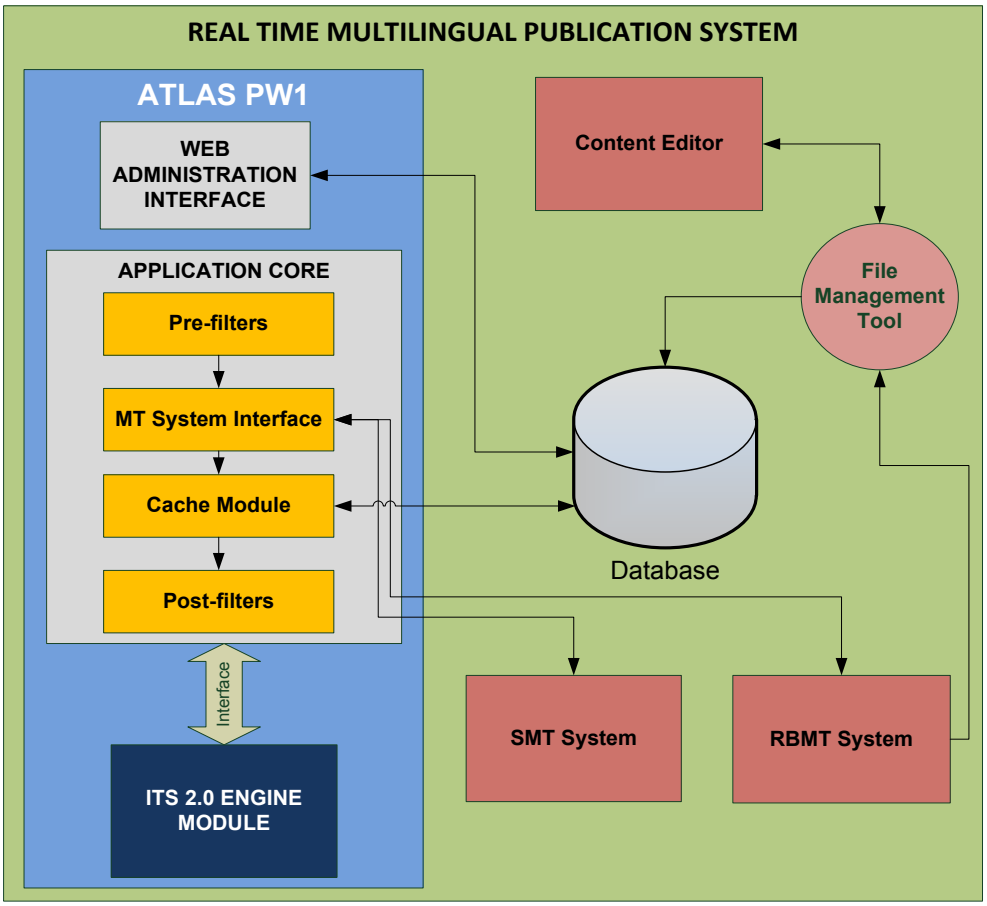

Figure 6.3: Module architecture

The next figure after zooming into ITS 2.0 Engine Module from the previous picture shows how ITS 2.0 is integrated in the System.

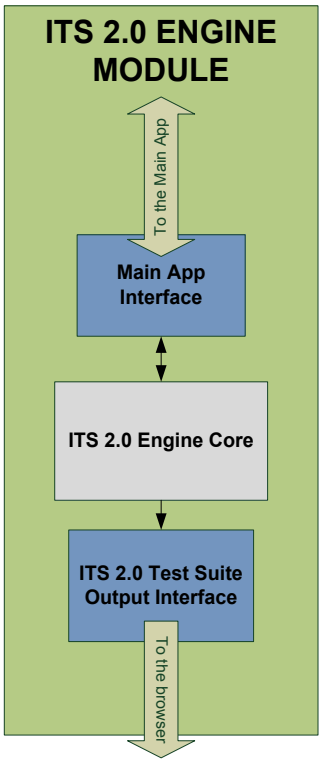

Figure 6.4: ITS 2.0 Module

As stated before, the above figure shows how the ITS 2.0 module is embedded and interconnected with the System. As shown in the figure, the ITS 2.0 module is an independent module of the System that is interfaced with the Main Application, the module has an engine sub-module in charge of processing all the metadata, and in the same way it has an interface to generate and display the output files of the Test Suite in a browser.

FUNCTIONAL ANALYSIS

This section explains how must the behaviour of the System be when interacting with the selected metadata, given the workflow description of the System (figure X, section 6.1), and specification of the before-mentioned metadata (see section 6.2.1). Individually for each metadata a brief description of its functionality in the System and examples are provided.

TRANSLATE

The non-translatable content is marked as a constant and will not be translated whether it pertains to text nodes or attributes, the latter only via global rules.

In the following example, the text *"World Wide Web Consortium"* will be left un-translated alongside with all the attributes that by default are non-translatable.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag test: Default</title>
  </head>
  <body>
    <p>The <span translate=no>World Wide Web Consortium</span> is
      making the World Wide Web worldwide!</p>
  </body>
</html>
```

LOCALIZATION NOTE

The system captures the text and type of the note that is conveyed to the Content Editor.

In the following example, after the document is processed and sent to the Content Editor, the Editors will see the advice note as an alert embedded with the sentence *"This is a motherboard"*.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>LocNote test: Default</title>
  </head>
  <body>
    <p>This is a <span its-loc-note="Check with terminology engineer" its-loc-note-
type=alert>motherboard</span>.</p>
  </body>
</html>
```

LANGUAGE INFORMATION

The system uses the language information of the different nodes to automatically detect the source language and updates the ***lang*** attribute of the output.

In the following example, after the document is processed the value of the ***lang*** attribute will be changed to the specific value for the target language, the mentioned value can be mapped according to a personal configuration if needed.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Lang Info test: Default</title>
  </head>
  <body>
    <p>Some text.</p>
  </body>
</html>
```

DOMAIN

The different domain values are mapped depending on the MT System used. The System allows using several different domains on the same document.

In the following example, after the document is processed, the content of the element p (paragraph) will be sent to the MT System to be translated using the sports vocabulary of the glossary, on the other hand the content of the span will use vocabulary related to laws, the rest of the translatable content such as the title

will be machine translated using the default vocabulary of the MT System.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <meta name="description" content="SPORTS">
    <meta name="x-mykeywords" content="LAW">
    <script type="application/its+xml">
<its:rules xmlns:its="http://www.w3.org/2005/11/its"
xmlns:h="http://www.w3.org/1999/xhtml" version="2.0">
<its:domainRule selector="/h:html/h:p"
domainPointer="/h:html/h:head/h:meta[@name='description']/@content"/>
<its:domainRule selector="/h:html/h:p/h:span"
domainPointer="/h:html/h:head/h:meta[@name='x-mykeywords']/@content"/>
</its:rules>
    </script>
    <title>Domain test: Default</title>
  </head>
  <body>
    <p>Some text about sports and <span>some related to laws<span>.</p>
  </body>
</html>
```

PROVENANCE

The information provided by the MT Systems and by the editors via the Content Editor, is added to the nodes of the document in order to provide information to the user about who intervened in the translation and revision processes.

In the following example, the information provided by the MT System after the translation process and by the Content Editor after the revision process is stored on the Provenance Records indicating that the translation process was performed by the Online MT System of the ACME company and a subsequent revision was performed by John Smith with the Online Revision Tool of the XYZ Corporation.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script id=pr1 type=aplication/its+xml>
      <its:provenanceRecords xml:id="pr2" xmlns:its="http://www.w3.org/2005/11/its"
version="2.0">
        <its:provenanceRecord toolRef="http://www.onlinemts.com/mtengine"
org="ACME"/>
        <its:provenanceRecord revPerson="John Smith"
toolRef="http://www.onlinerevt.com/ce" org="XYZCORP"/>
      </its:provenanceRecords>
    </script>
  </head>
  <body>
    <p its-provenance-records-ref="#pr1">This paragraph was translated from the
machine and subsequently revised.</p>
  </body>
</html>
```

LOCALIZATION QUALITY ISSUE

The information regarding the localization quality can be added in the original content by the user or provided by the reviser via the Content Editor. Later this information can be used in scenarios such as, MT developers using it to improve the MT System engine or the user using it to warn the revisers of revision errors.

In the following example, after the revision process the user detects two issues related to the word *"c'es"*,

then adds both localization quality issues to the content so that they will be conveyed to the Content Editor to warn the revisers about them in order to be corrected.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script src=qaissues.js type=text/javascript></script>
    <script type=application/its+xml id=lq1>
      <its:locQualityIssues xml:id="lq1" xmlns:its="http://www.w3.org/2005/11/its">
        <its:locQualityIssue
          locQualityIssueType="misspelling"
          locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
          locQualityIssueSeverity="50"/>
        <its:locQualityIssue
          locQualityIssueType="typographical"
          locQualityIssueComment="Sentence without capitalization"
          locQualityIssueSeverity="30"/>
      </its:locQualityIssues>
    </script>
    <style type=text/css>.qaissue { background-color: yellow; } </style>
  </head>
  <body onload=addqaissueattrs()>
    <p>
      <span its-loc-quality-issues-ref=#lq1>c'es</span> le contenu.</p>
  </body>
</html>
```

6.3 ITS 2.0 implementation

The necessary changes and the implementation efforts performed on the adaptation of the System to be ITS 2.0 compliant, can be classified into three main steps:

1) Development of the ITS 2.0 metadata rules engine.
2) Integration of the System with the ITS 2.0 metadata rules engine.
3) Adaptation of the System to allow ITS 2.0 configurations and projects.

In the following sub-sections each step is explained in more detail.

ITS METADATA RULES ENGINE

To process the ITS 2.0 metadata an engine has been developed, in collaboration with Daedalus, taking into account the particulars, requirements, features and restrictions inherent to each metadata such as precedence, inheritance, overriding, defaults, local and global implementation, external rules, stand-off markup, IRIs and parameters. The System can only process both HTML5 and XML.

With metadata such as *Localization Note*, *Provenance* and *Localization Quality Issue*, a new concept of encoding information is introduced. These metadata allow information to travel between the different modules of the System. The before-mentioned metadata will be transformed from the original markup to a brand new format called ***Special Plain Text (SPT)*** and the other way around; the main reason is because the Translation Memories of the MT Systems do not deal with markup, just with plain text. The SPT format matches the following pattern.

\[@@metadata_attr=(“|’)?.*?(“|’)?(\&\&metadata_attr=(“|’)?.*?(“|’)?)*(@@#@@metadata_attr=(“|’)?.*?(“|’)?(\&\&metadata_attr=(“|’)?.*?(“|’)?)*)*@@\]

According to the previous pattern, SPT, is able to wrap one or more records of the same metadata from one node of the document and even from different data categories, under the same data structure. The following example includes *Localization Note* and *Localization Quality Issue* within the same SPT.

[@@its-loc-note="Check with terminology engineer"&&its-loc-note-type=alert@@#@@its-loc-quality-issue-comment="should be 'quality'"&&its-loc-quality-issue-severity=50&&its-loc-quality-issue-type=spelling@@]

The MT Systems will skip the translation of the SPTs. The SPTs will be uploaded into the ***Content Editor*** along with the rest of the content to edit, and eventually, depending on the case, it will be saved in the translation memory.

TRANSLATE

After correctly processing this metadata, the System will block the content of those elements and attributes considered non-translatable by wrapping the content with own-proprietary tags.

For the content of the elements of the document, the selected tag is a comment tag with the following format (bold red): **<!--PMT_no_trans scope="text"-->Do not translate this text<!--/PMT_no_trans-->**. For the content of the attributes the tag selected has the following format (bold brown): ****. ANT stands for attribute non-translatable.

DOMAIN

The System is capable of dealing with various different domains per document, that is, make translation requests to the MT System using a list of domains for the whole document or on specific sections of the document. To achieve that purpose the System wraps with an own-proprietary tag all the nodes that belong to a pre-defined section and assigns to it the list of domains that applies to that section:

** <!--PMT_section domains="ECON"-->**Some text about economy and **<!--/PMT_section--><!--PMT_section domains="LAW"-->**some related to laws**<!--/PMT_section-->. **

The possibility of using different domains in sub-sections inside a section is out of the scope of the implementation.

LANGUAGE INFORMATION

The System updates the ***lang*** value of the nodes depending on the rules after any translation request. The System can be configured to map those values if necessary.

LOCALIZATION NOTE

As stated before the information provided by the *Localization Note* data category needs to be transformed in order to be sent to the ***Content Editor,*** to such end the ***SPT*** is used as a means to convey such information.

The process of encoding the information is comprised of four steps. Here is an example to illustrate the mechanism.

1. The System gets the original content.

** This is a motherboard. **

2. The System converts the original content into the SPT and adds it to the text node with a whitespace.

** This is a [@@its-loc-note="Check with terminology engineer"&&its-loc-note-type=alert@@] motherboard. **

3. The System receives the translated content from the MT System.

** Esto es un [@@its-loc-note="Check with terminology engineer"&&its-loc-note-type=alert@@] placa madre. **

4. The System removes the markup no longer needed. The System can be configured to keep the markup.

** Esto es una placa base. **

The same process could be carried out with global rules.

PROVENANCE

As stated before, the information provided by the *Provenance* data category needs to be transformed from the ***SPT*** to markup. The information presented by this metadata is provided by both MT System and Content Editor.

The System with this metadata works the other way around compared to *Localization Note* and inverts the mechanism of conversion from SPT to markup. The process of decoding the information is comprised of four steps. Here is an example to illustrate the mechanism.

1. The System gets the original content.

** Original text. **

2. The System adds in the first place the relevant information regarding the MT System used in the translation process. The information details with respect to the MT System is configured in the System's configuration

** El texto original ha sido traducido. **

3. The System then adds the relevant information related to the revision process in a SPT format.

** [@@its-rev-person=" Tommy Atkins"&&its-rev-org="ACME"@@] El texto original ha sido revisado. **

4. Finally the System converts the SPT into markup.

** El texto original ha sido revisado. **

There are some scenarios where a concrete attribute has to be added to some nodes with different values, for

example, a revision of a text previously revised. To allow such cases local *standoff markup* is provided. When the System detects more than one occurrence for the same data category it encodes several provenance records as follows:

<its:provenanceRecords xml:id="pr1" xmlns:its="http://www.w3.org/2005/11/its" version="2.0"><its:provenanceRecord revPerson="Tommy Atkins" revOrgRef="http://www.linguaserve.com/" /><its:provenanceRecord revPerson="John Smith" revOrgRef=" http://www.linguaserve.com/" /><its:provenanceRecords>

** This paragraph was revised twice. **

LOCALIZATION QUALITY ISSUE

As stated before, the information provided by the *Localization Quality Issue* data category needs to be transformed in order to be sent to the ***Content Editor,*** to such end the ***SPT*** is used as a means to convey such information. The ***Content Editor*** allows inserting this information to be displayed into the translated content.

The System with this metadata works, as explained above, both ways: from the original content to the translated content and vice versa. The process of encoding/decoding the information is comprised of five steps. Here is an example to illustrate the mechanism.

1. After a certain translation request, the MT System, for some reason, leaves one word un-translated.

** The calidad of the translation is poor. **

2. Afterwards the content author detects it and creates a localization quality issue.

** La calidad de la traducción es pobre. **

3. The System gets the original content and converts the original content into the SPT and adds it to the text node with a whitespace.

** La calidad[@@its-loc-quality-issue-comment=" calidad should be 'quality'"&&its-loc-quality-issue-severity=60&&its-loc-quality-issue-type=untranslated@@] de la traducción es pobre. **

4. The System receives the content revised with the localization quality issue enabled flag deactivated.

** The quality[@@its-loc-quality-issue-comment=" calidad should be 'quality'"&&its-loc-quality-issue-severity=60&&its-loc-quality-issue-type=untranslated&&its-loc-quality-issue-enabled=no@@] of the translation is poor. **

5. Finally the System removes the mark-up no longer needed. The System can be configured to keep the mark-up.

** The quality of the translation is poor. **

There are some scenarios where a concrete attribute has to be added to some nodes with different values, for example, a word with two issues one related to misspelling and another related to typographical. To allow

such cases local *standoff mark-up* is provided. When the System detects more than one occurrence for the same data category it encodes several localization quality issues as follows:

<its:locQualityIssues xml:id="lq1" xmlns:its="http://www.w3.org/2005/11/its"> <its:locQualityIssue locQualityIssueType="misspelling" locQualityIssueComment="'c'es' is unknown. Could be 'c'est'" locQualityIssueSeverity="50"/><its:locQualityIssue locQualityIssueType="typographical" locQualityIssueComment="Sentence without capitalization" locQualityIssueSeverity="30"/></its:locQualityIssues>

c'es le contenu</p>

INTEGRATION OF THE SYSTEM WITH THE ITS 2.0 METADATA RULES ENGINE

The System's ITS 2.0 metadata rules engine is an independent module, and it is integrated with the System via an interface. This task has been achieved in collaboration with Daedalus. In order to be ITS 2.0 compliant, the System will process HTML5 and construct a DOM out of it. According to the ITS 2.0 general processing requirements the new workflow is as follows:

1. Downloading of the original content
2. Creation of the DOM associated to the document
3. Parsing of the elements and attributes of the DOM
4. Processing the metadata according to the behaviour expected from their specifications
5. Conversion to HTML5
6. Sending of the document to the MT System
7. Receiving of the translated document from the MT System
8. Creation of the DOM associated to the translated document
9. Parsing of the elements and attributes of the DOM
10. Processing the metadata according to the behaviour expected from their specifications
11. Conversion to HTML5

When the System detects an error during the DOM's construction stages (e.g.: mal-formed HTML5), it will return an internal server error (Error 500).

ADAPTATION OF THE SYSTEM TO ALLOW ITS 2.0 CONFIGURATIONS AND PROJECTS

Thanks to its new modular architecture, the System is capable of evolving in a faster and easier way, providing a more dynamic behaviour.

The modules management is handled by the System's core engine and can be configured via the System's administration console. On this new version, the System allows the System's administrator or the project managers:

- To activate and deactivate the ITS 2.0 module
- To manage and assign MT Systems per language pair
- To activate and deactivate pre-existing modules

To make this possible, the following adaptations were performed on the different modules:

WEB ADMINISTRATION MODIFICATIONS

- Possibility to map different modules for a given project
- Activation and deactivation of the removal of the metadata not needed in the output during the post-processing stage
- Configuration of the locale values
- Configuration per language pair to select different MT Systems

DATABASE MODIFICATIONS

- New columns and indexes on existing tables and new tables to store the information needed to configure the new functionalities

MT SYSTEM CONNECTION INTERFACE MODIFICATIONS

- Mapping of the connection client and translation parameters to the selected MT System
- Possibility to specify the appropriate domain of the translations, if the value is null or empty then the MT System will use the default value

CACHE SYSTEM MODIFICATIONS

- Creation of a sub-module that controls whether the references to external files related to the ITS process change their content or not, in order to add this information to the cache unique identifier

TEST-SUITE OUTPUT INTERFACE

- Creation of an interface with the ITS 2.0 engine that generates an output format Test-Suite file after the processing of any file, and also shows it on the browser. This functionality is configurable from the web administration interface

6.4 Contribution to the Showcase

This showcase is made in collaboration with the Spanish Tax Agency – AEAT (http://www.agenciatributaria.es). For this showcase, the Spanish Tax Agency is in the process of migrating their website to HTML5, and will add a substantial amount of the selected metadata throughout the website via scripts and CMS with HTML5 plus ITS 2.0 validation.

The modifications made on Linguaserve's Online System will process the metadata inserted into AEAT's website, providing all the functionalities mentioned on the previous sections.

The scenarios for the showcase are:

- *Production environment (upon approval of the client)*
    - LucySoftware RBMT System with the language pair Spanish to English
- *Pre-production environment*
    - LucySoftware RBMT System with the language pair Spanish to French
    - LucySoftware RBMT System with the language pair Spanish to German
    - DCU SMT System with the language pair Spanish to English

For the showcase, due to restrictions requested by the client, it is not possible to use ITS 2.0 compliant MT Systems. The reason is that one of the requirements of the translations for this client is that all the texts must be completely revised, i.e. coming out of the Translation Memory, because of the translation quality of the MT System is not completely accurate and the texts of the AEAT are usually very sensitive. In case the content is not completely revised it will be published in the original language, in addition, the documents must be divided into sections such as the header, lateral menus, main body, or footer, as a means to return to the user the least content possible in the original language.

Given those restrictions, the initial idea was to divide the document into well-formed HTML5 so that an 2.0 compliant MT System ITS would be able to process metadata, such as *translate* with rules, e.g. doing something like this:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset=utf-8>
 <title>Example sections</title>
 <script type=application/its+xml>
 <its:rules xmlns:its="http://www.w3.org/2005/11/its"  xmlns:h="http://www.w3.org/1999/xhtml"
version="2.0">
 <its:translateRule selector="/h:html/h:body/h:div/h:span" translate="no"/>
 </its:rules>
 </script>
</head>
<body>
 <div id="header">
     Header content <span>do not translate</span>.
 </div>
 <div id="main">
 <p>
   Main content <span>do not translate</span>.
 </p>
 </div>
 <div id="footer">
 <p>
   Footer content <span>do not translate</span>.
 </p>
 </div>
</body>
</html>
```

Given the previous example the System would create the three sections such as the following, with the html, head and body elements:

Section 1:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset=utf-8>
 <title>Example sections</title>
 <script type=application/its+xml>
 <its:rules xmlns:its="http://www.w3.org/2005/11/its"  xmlns:h="http://www.w3.org/1999/xhtml"
version="2.0">
 <its:translateRule selector="/h:html/h:body/h:div/h:span" translate="no"/>
 </its:rules>
 </script>
</head>
<body>
 <div id="header">
     Header content <span>do not translate</span>.
 </div>
</body>
</html>
```

Section 2:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset=utf-8>
 <title>Example sections</title>
 <script type=application/its+xml>
 <its:rules xmlns:its="http://www.w3.org/2005/11/its"  xmlns:h="http://www.w3.org/1999/xhtml"
version="2.0">
 <its:translateRule selector="/h:html/h:body/h:div/h:span" translate="no"/>
 </its:rules>
 </script>
</head>
<body>
 <div id="main">
 <p>
   Main content <span>do not translate</span>.
 </p>
 </div>
</body>
</html>
```

Section 3:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset=utf-8>
 <title>Example sections</title>
 <script type=application/its+xml>
 <its:rules xmlns:its="http://www.w3.org/2005/11/its"  xmlns:h="http://www.w3.org/1999/xhtml"
version="2.0">
 <its:translateRule selector="/h:html/h:body/h:div/h:span" translate="no"/>
 </its:rules>
 </script>
</head>
<body>
 <div id="footer">
 <p>
   Footer content <span>do not translate</span>.
 </p>
 </div>
</body>
```

```
</html>
```

But that section division is not that easy and is strongly dependent of the document structure, e.g.:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
 <div id="main">
 <div id="header">
  </div>
 <div id="body">
  <div id="left_menu">
   </div>
   <div id="content">
   </div>
  <div id="right_menu">
   </div>
  </div>
 <div id="footer">
  </div>
</div>
</body>
</html>
```

With this structure the division into sections is not possible because our System cannot manage subsections, and otherwise it would end up sending to the MT System mal-formed HTML5, which would entail problems for the MT System trying to create a DOM out of it.

Since this is a client's requirement, it is not possible to turn it back and, besides, that would be an involution of the System, which is not acceptable.

So to summarise, it is not possible to connect the System to an ITS 2.0 compliant MT Engine version because of the HTML5 integrity sent to the MT System is not guaranteed, and the System can manage all the selected metadata for the showcase by itself.

7. QUICK GUIDELINE TO THE APPLIED ITS 2.0 TAGGING

ITS 2.0 Data Category	Behaviour	Implemented in
Translate	Non-translatable content is marked as a constant and is not translated, whether it pertains to text nodes or attributes; the latter only with global rules.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1, DCU's Statistical MT System MaTrEx, and LucySoftware's Rule-based MT
Localization Note	The system captures the text and type of the note that is conveyed to the Content Editor so as to help the editor in the localization process.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1
Language Information	The system uses the language information of the different nodes to automatically detect the source language and update the *lang* attribute of the output.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1, DCU's Statistical MT System MaTrEx, and LucySoftware's Rule-based MT
Domain	The different domain values are mapped depending on the MT System used, and only one per document will be permitted when it comes to MT Systems.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1, DCU's Statistical MT System MaTrEx, and LucySoftware's Rule-based MT
Provenance	The information provided by the MT Systems and by the editors via the Content Editor is added to the nodes of the document in order to provide information to the user.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1
Localization Quality Issue	The information regarding the quality of the localization can be added to the original content by the user or be provided by the reviser via the Content Editor. Later, this information can be used by the MT developers to improve the MT System core, for instance.	Linguaserve's Real Time Multilingual Publication System ATLAS PW1

Locale Filter	The Locale Filter data category is used to specify that a node is only applicable to certain locales (useful in localization)	DCU's Statistical MT System MaTrEx
MT Confidence	The MT Confidence data category is used to communicate the confidence in the quality of the translation (output by the MaTrEx system).	DCU's Statistical MT System MaTrEx

8. REFERENCES

HTML5

Ian Hickson HTML5 – A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft 29 March 2012. Available at http://www.w3.org/TR/html5/.

ITS 2.0 standard

Lastest ITS 2.0 WD.

Language Technology

Hans Uszkoreit Language Technology A First Overview.

Linguistics

Linguistic Annotation.

MT Systems

Statistical Machine Translation.

Miles Osborne, MT History and Rule-based systems.

Multilingual Web

http://www.w3.org/International/multilingualweb/lt/.

http://www.multilingualweb.eu/.

Provenance

Provenance data model.