D4.1.1: LUCY MODIFICATION

Thomas Ruedesheim

Distribution: Public

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

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	4.1.1
Deliverable title:	Lucy Modification
Dissemination level:	PU
Contractual date of delivery:	31st January 2013
Actual date of delivery:	15th February 2013
Author(s):	Thomas Ruedesheim
Participants:	Lucy Software
Internal Reviewer:	Lucy Software
Workpackage:	WP4
Task Responsible:	Thomas Ruedesheim
Workpackage Leader:	Pedro L. Díez Orzas

Revision History

Revision	Date	Author	Organization	Description
1	13/02/2013	Thomas Ruedesheim	Lucy Software	Final version
2	21/06/2013	Thomas Ruedesheim	Lucy Software	Added technical information on modification to process ITS 2.0
3	24/04/2013	Felix Sasaki	DFKI	Minor revision (link fixes etc.)

CONTENTS

1 LUCY MODIFICATION

1.1 Description

The Lucy LT Engine is a rule-based machine translation (RBMT) system. It consists of two main components, the Text Handling module and the MT Kernel. The Text Handling module is responsible for the conversion of formatted input (text, documents, and web pages) into a file of input segments (plain text) and the format information. The Kernel performs the translation process, yielding in a file of output segments (plain text). Finally, the Text Handling module produces formatted output using the format information that has been extracted before. The Text Handling module can deal with various formats, including HTML, XML, RTF, and MS Office® (Word, Excel, and PowerPoint).

We have modified the HTML sub-module of the Text Handling module to handle HTML5 (HTML representation) and the ITS 2.0 data categories Translate and Domain. There are also routines handling the HTML5 “lang” attribute (interpreted as ITS 2.0 “Language Information”).

The Kernel currently supports 34 translation directions for European languages.

1.2 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.

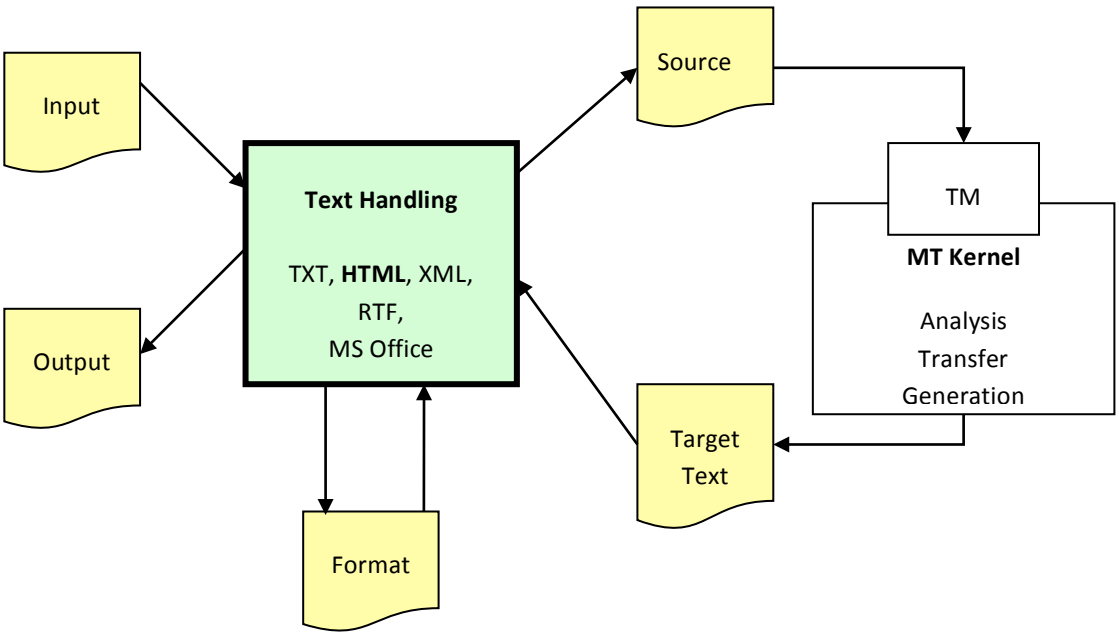


Figure 4.1: LT Engine Architecture - Workflow

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).

1.3 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.

1.3.1 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/its20/

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/its20/#trans-datacat

DOMAIN

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

More info on: http://www.w3.org/TR/its20/#domain

1.3.2 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 that 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

1.3.3 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 specification http://www.w3.org/TR/its20/#datacategories-defaults-etc.

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.

1.4 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 ITS 2.0 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.

1.5 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.

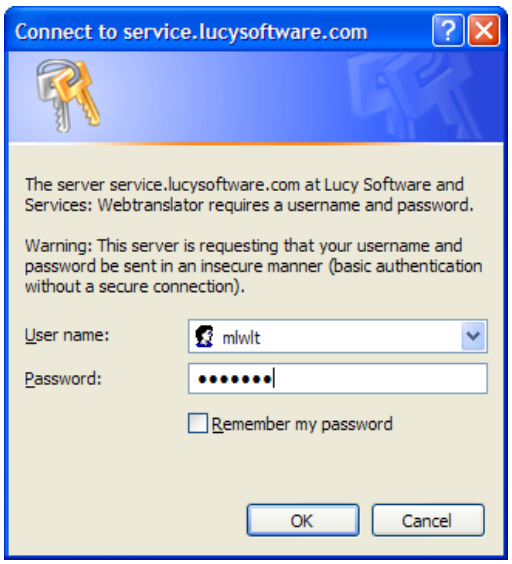
2 USER GUIDE

Link to prototype or downloadable: http://service.lucysoftware.com:8080/dts

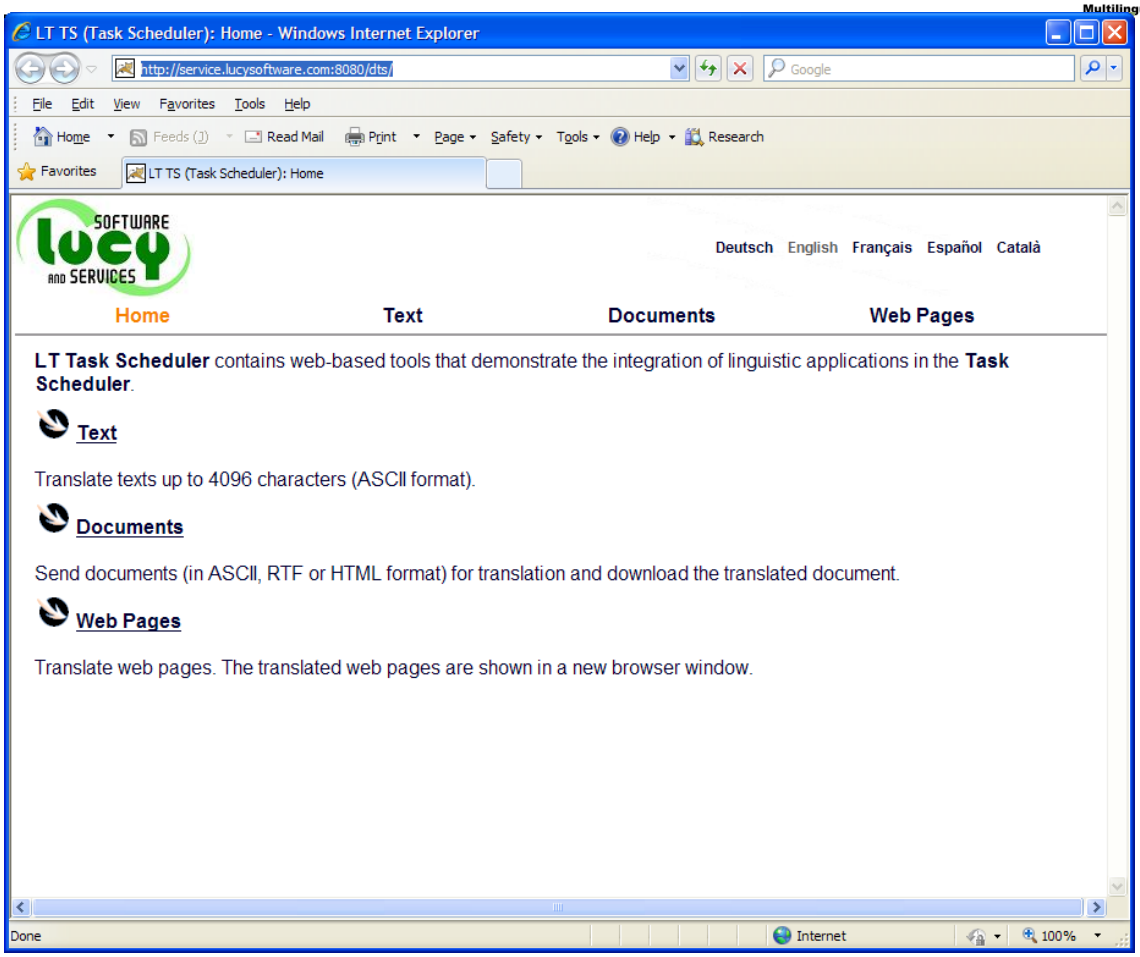
The latest LT Engine (EX_6.11_ITS.004) is running behind an LT TaskScheduler installation at Lucy. There are several ways for accessing the translation engine:

- LT AnyWhere desktop client
- Demo web client for LT TaskScheduler's web server
- Web service client for RESTful web service AutoTranslateRS

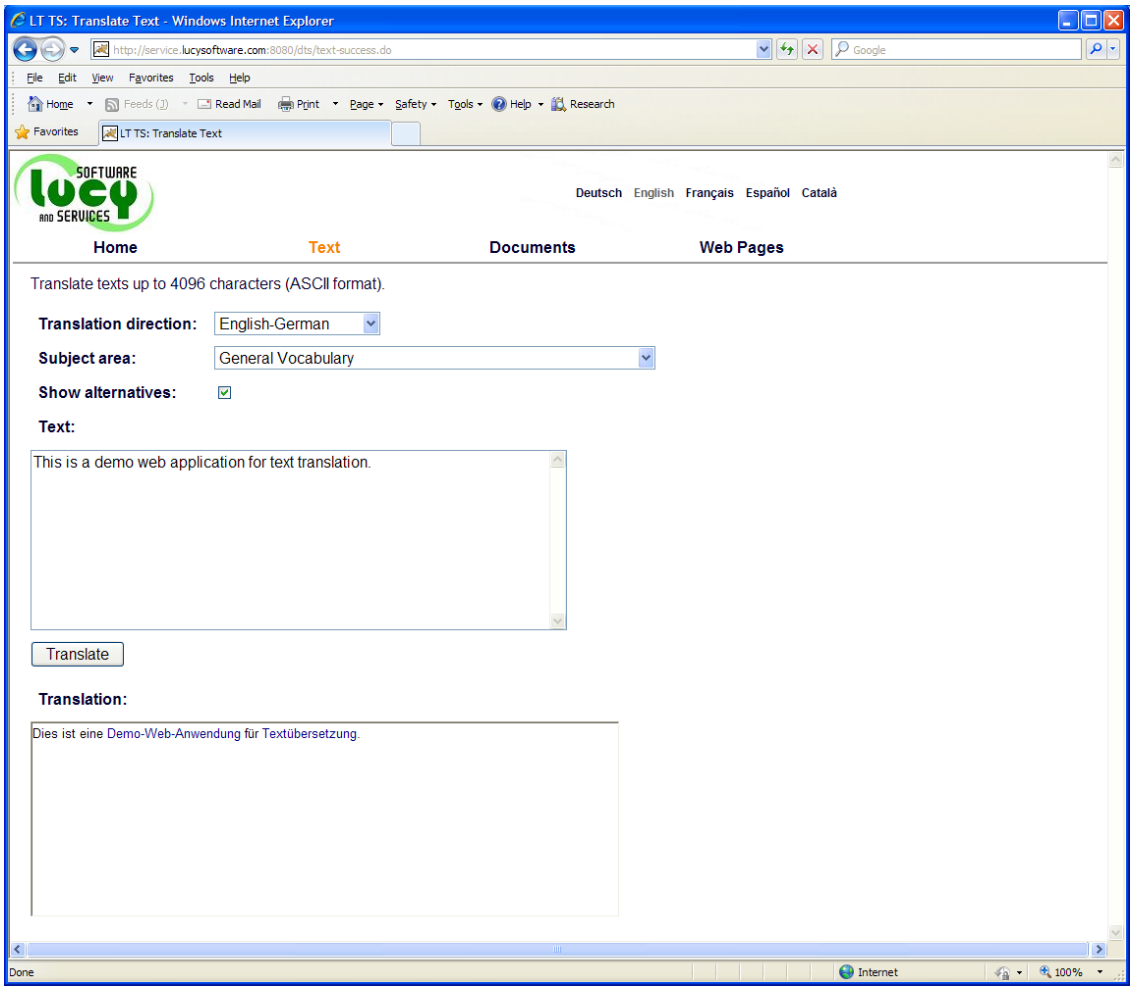
For testing purposes the second alternative is best suited, since it requires a web browser only. The service URL is http://service.lucysoftware.com:8080/dts. Use the following credentials: "mlwlt" / "ltweb11"



The next screen offers translation of text, documents, or web pages:


LT TS (Task Scheduler): Home - Windows Internet Explorer
http://service.lucysoftware.com:8080/dts/
File Edit View Favorites Tools Help
Home Feeds (J) Read Mail Print Page Safety Tools Help Research
Favorites
LT TS (Task Scheduler): Home
SOFTWARE lucy AND SERVICES
Deutsch English Français Español Català
Home Text Documents Web Pages
LT Task Scheduler contains web-based tools that demonstrate the integration of linguistic applications in the Task Scheduler.
Text
Translate texts up to 4096 characters (ASCII format).
Documents
Send documents (in ASCII, RTF or HTML format) for translation and download the translated document.
Web Pages
Translate web pages. The translated web pages are shown in a new browser window.
Done
Internet
100%

Click on “Text” and choose a translation direction (e.g. English-German) and un-check “Show alternatives”. Then enter some text in the “Text” input field and press the translate button. The translated text will appear in the “Translation” text field.



You may also enter HTML formatted text in the text field. ITS rules must be in-line.