S

Initial Survey on the Availability of Translation Memory Tools

Country Component	Sri Lanka	**Report no.**	
Phase no.	**1.2**	**Report Ref no.**	

Prepared By: **Mr. Ruwan Asanka Wasala Designation: Research Assistant**

Project Leader: **Dr. Ruvan Weerasinghe** **Signature:** _____

Date: _____

TABLE OF CONTENTS

Initial Survey on the Availability of Translation Memory Tools

Ruwan Asanka Wasala
Language Technology Research Laboratory,
University of Colombo School of Computing,
35, Reid Avenue,
Colombo 07,
Sri Lanka.
raw@ucsc.cmb.ac.lk

Abstract

This report begins with a brief introduction to the Translation Memory (TM) systems, followed by an explanation of the Translation Memory Process. It then presents a feature-comparison of most popular commercial translation memory systems available today. This research is mainly focused on reporting the state of the art of available Open-source translation memory systems. Followed by the description of Open-source TM systems, the next section summarizes open language standards such as XLIFF, TMX, TBX and SRX that are widely used in computer aided translation tools. Finally the report concludes with a set of features to be included in a TM system to be developed. Findings of this research suggests that Open-source TM system "Transolution" is the most suitable system which can conveniently integrate the suggested features.

1. Introduction

Present huge demand for translations cannot be met with due to lack of efficient and competent translators. Translation is a complex activity requiring high level of skill, and it is a very expensive process [3]. So requirement of finding cheaper and faster solutions to international exchange of information has resulted in translation tools to be designed to assist human translators.

The enormous boost in information and communication technology combined with the modern world's sophisticated multilingual communication requirements ask for fast and automatic translation. As a result, today many computer aided translation tools such as multilingual dictionaries, grammar and spell checkers, terminology managers, concordancers, online bilingual texts, machine translation (MT) systems, translation memory (TM) managers etc [2][3] have become available to translators (A list of available proprietary and free CAT tools are presented in [2],[4] and [5]). These tools support and facilitate human translators in the translation process. The translation process itself has been inevitably affected by these tools.

Among the above-mentioned tools, MT systems and TM systems are considered to be more complicated and advanced CAT tools. A Machine Translation system is software that could fully replace a human translator. MT systems often use advanced artificial intelligence technologies and statistical computational methods along with rich linguistic data (such as grammatical rules, comprehensive dictionaries) to translate the source text into suitable grammatically correct target sentences. Thus, MT is a computationally expensive task.

On the other hand, a TM system is used as a translator's aid. A translation memory (TM) is a type of database that stores source and target language pairs of text segments that can be reused in new translation projects. TMs are designed to increase the efficiency of human translators' work by means of suggesting previously translated text fragments for similar text fragments that are being translated by avoiding keying-in previously translated text fragments repeatedly. TMs are used in automatically translating identical text segments or excluding multiple translations of similar text fragments. TMs are extremely useful in maintaining consistency with previous translations. TMs can be classified into different types based on information stored and the retrieval methods. This survey is an attempt to report the State of the Art of translation Memory tools while reporting the suggestions for developing a multilingual Translation Memory framework that supports common translation memory exchange formats.

1.1 Definition of Translation Memory

Expert Advisory Group on Language Engineering Standards (EAGLES) Evaluation Working Group's has defined a translation memory (TM) system as [1],

> *"a multilingual text archive containing (segmented, aligned, parsed and classified) multilingual texts, allowing storage and retrieval of aligned multilingual text segments against various search conditions."*

1.2 Difference between Translation Memory and Machine Translation Systems

MT systems use advanced natural language processing techniques to translate source text into grammatically correct target sentences. Normally, MT systems are capable of translating entire document into target language without any human interaction. For the same reason, the accuracy and the quality of the translation is rather skeptical. However, TM systems are not sophisticated in nature. A TM system only helps human translator by suggesting previously translated content when they appear in a new document. Once a translation has been performed using a TM manager, not only is there a glossary of terms stored for future recall, but also text fragments (such as phrases or sentences) will also be stored, allowing quicker future translation of content. In this way, TM provides efficient and effective methodology for managing, retrieving and reusing previously translated text. TMs achieve this by matching up the original source document text fragments with its database (which contains previously translated source and resulting target text segments) with the updated or revised document through 'exact' and 'fuzzy' matching. As such one can not expect fully automatically translated document as the output of a TM manager, although such system will expedite the translation process.

2. Translation Memory Process

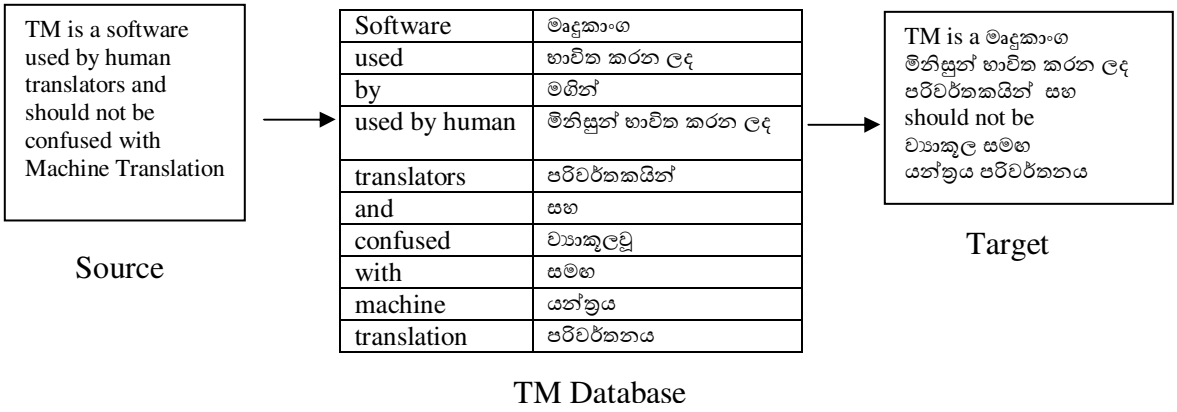

Figure 1. TM Process

The key processes of a TM tool include text segmentation, text alignment, indexing, search and match retrieval [7][1]. Generally, the text segmentation and alignment processes are carried out before indexing the text. In the text segmentation process, source and target sentences are segmented into various translation units. Usually, the basic unit of a text in a TM system is a sentence; this unit might even be a paragraph, a phrase, a sentence, a sentence fragment, or a word, but in some systems the user can also define what the unit will be [6]. Once the segmentation process is completed, the text segments (pairs of translation units) will be aligned. The alignment involves matching the source texts by aligning matching target text segments. Then the translational units will be indexed, and stored in the terminology database along with other useful attributes such as document creation date, author, the client, project ID, domain etc [7]. This approach is used in popular commercial TM systems including TRADOS, SDLX, STAR Tansit and Deja Vu, among others [7] .

Unlike conventional TM system, some other tools avoid the text segmentation process by employing a different approach, called the full-text approach or reference model [6][2][7]. These systems store content as full bitexts (full source- and target-text pairs) and index them in the TM database using the character-string-in-bitext (CSB) technique. Once the bitexts are in the database, they are aligned at paragraph level. This approach is used in commercial TM software such as TransSearch, LogiTerm, and MultiTrans [7][8]. Advantages of this recent method over traditional method include faster creation of a large TM databases and the retention of content for any match found and suggested to the user.

Considering the issues related to database maintenance and security, TM systems have been developed without TM databases which store previously translated material. Such systems usually create 'virtual' translation memory on the fly by associating the translated files that reside in any directory on one's computer [7].

The core element of every TM system is the match retrieval process. The main task of a translation memory is to match text fragments in the current document to those stored in the TM system's database and retrieve the suitable translated text fragments (see Figure 1) from it. In the search process, there may be situations where exact matches are very rare; in such situations the system must also find text fragments (phrases, sentences, words) similar to the current one. As such, TM system's search process can either return an exact or a fuzzy (proximity) match [1][7]. In Exact matching, the TM system pairs text segments in a source

document that match the original source text exactly. But, any text in the document that does not exactly match the original will not be translated. Fuzzy matching, will however find segments that are very similar to the original text segment and suggest the original translation. Though fuzzy matching is useful, in certain situations translations retrieved by fuzzy matching can be inappropriate in the context. Therefore, manual-post editing by the translator becomes essential.

Once the search process is over for each text fragment, the TM provides the corresponding translation stored in the database. In order to optimize this search and retrieval process, TM systems usually stores and indexes previously translated content in an organized way. This enables the retrieval of best matching translation units within a very short period. The efficiency of a TM system depends on the retrieval of all available exact or fuzzy matches for a source segment (match recall) and the accuracy of the exact or fuzzy matches for the source segment (match precision) in context [7].

Mainly, two matching techniques have been reported to use in TM systems. The first technique, "character-string based method" tries to recognize matches not only at segment level but also in sub-parts of the segments. The second technique is known as "linguistically enhanced matching", and this technique make use of natural language processing techniques and resources to analyse sentences, to identify and separate different syntactic and semantic text chunks [1][7][8]. Furthermore, some systems employ Part-of-Speech tagging and other grammatical annotations along with advanced algorithms such as statistical or neural network based approaches to pair the best translation units. The latter technique was employed more recently and is reported to produce improved results in terms of both precision and recall. However, one of the crucial disadvantage of the systems those use linguistically enhanced matching technique is that these systems are language depended.

After the search and retrieval process, the translator can modify the remaining text segments that reflect the changes between the source and target texts without retranslating the entire document.

3. Commercial TM Systems

In this section different features of popular commercial TM systems have been introduced. Most popular commercial TM systems as reported in "TM Suvery 2006" are listed below in the descending order of popularity [7].

- TRADOS
- Déjà Vu
- Wordfast
- STAR Transit
- MultiTrans
- PASSOLO
- CatsCradle

For the comparison, features such as translation environment, supported file formats and open language standards such as translation memory exchange formats (TMX), computed statistics, matching methods and capability of handling special elements are considered in this study. Most commercial systems come with an in-built translation environment (e.g. Trados, Déjà Vu etc), while other TM systems (such as Wordfast, Trados) provide external interfaces for the existing word processing and desktop publishing software such as Microsoft Word, Adobe PageMaker etc via plug-ins and additional toolbars [2]. Few commercial systems offer both interfaces. Most of the commercial TM systems are superior to open-source TM systems due to their ability to pre/post-analyse documents and generate various reports containing useful statistics (e.g. word count, estimating amount of repetition in the text) [1][8]. Commercial TM systems support large number of input file formats such as text, Corel WordPerfect, WordPro and AmiPro, Microsoft Word documents (DOC and RTF), OpenOffice, OpenDocument (ODF), OpenDocument Text (.odt), OpenDocument Spreadsheet (.ods), OpenDocument Presentation (.odp), help files, resource files (RC), Microsoft Excel and PowerPoint, as well as tagged text file formats including DocBook, XML, SGML, HTML, ASP, JSP and QuarkXPress. Other file types include Adobe Frame, PageMaker, Interleaf, XGate, Gettext PO and C / C++, Java/VB, or other source code and binary files such as INI, EXE, OCX, DLL [2]. Moreover, in some systems users can define custom-made filters for other document types too (e.g. Star Transit). Another important feature is the number of languages supported by the system. Commercial TM systems support large number of languages. Specially developed algorithms and linguistic rules built into these systems deal with complexities unique to different languages. Some systems support over 100 languages. Furthermore, commercial TM systems consist of additional features such as in-built spell checkers and terminology databases. It is worthy to mention some advanced features built on to some of the commercial systems such as ability to access and share TM databases or resource in network environments [6][1]. Majority of commercial TM systems are developed for Windows. A good comparison of commercial TM systems and a comprehensive review is given in [2] and [28].

4. Open-source TM Systems

It is worth noticing the increasing interest and the trend in developing Open-source TM (OS-TM) software these days. Most of the open-source TM systems lack some advanced features offered by commercial products, but there exist working solutions. Open source TM software would definitely challenge the available proprietary TM tools within the next few years to come. This study was mainly focused on reporting the state of the art of OS-TM systems. Among OS-TM systems, the most successful projects are described here. These systems include, OmegaT, Transolution, ForeignDesk, Open Language Tools, OOxlate. It is also noteworthy to mention about recent OS-TM software developing initiatives such as opentm [10], to deliver multi-platform, open-source, full-featured, enterprise-grade translation memory tool.

4.1 OmegaT

Omega T is a free and Open Source multiplatform translation memory program released under GNU General Public License. OmegaT is written in Java. OmegaT was originally written by Keith Godfrey in 2000, however, now it is being developed by informal, international group of volunteers led by *Didier Brie* [11].

Features of OmegaT includes standalone working environment (graphics user interface), user customizable flexible segmentation (using regular expressions), fuzzy matching, match propagation, ability to work with multiple translation memories and projects, ability to use external glossaries, Unicode (UTF-8) support, Support for right-to-left languages and compatibility with other translation memory applications though improved TMX[1] support. Latest version (1.7.3) of OmegaT supports document formats including, XHTML, HTML Microsoft Office 2007 XML, OpenOffice.org/StarOffice (OpenDocument Format), Portable Object (PO) files, Java Properties files, XLIFF[2], DocBook, MediaWiki (Wikipedia) and Plain text. [11][12]. Moreover, a range of utilities and third-party software are available to enhance the functionality of OmegaT. Among these utilities dictionaries, alignment tools, text segmentation tools and file format translators are found to be very useful.

OmegaT comes with a comprehensive on-screen user manual and a brief "Instant Start" tutorial. OmegaT application, it's documentation, and OmegaT website have been translated into several languages by volunteers.

4.2 Translolution

Transolution (formerly known as *EvilTrans*) is an open-source platform independent computer aided translation suite written entirely in the Python language by *Frederik Corneliusson*. Transolution is released under GNU General Public License and its features include usability for both software and documentation, sentence segmentation, tag protection, interactive translation memory, and XLIFF format support [13].

Transolution is developed in a highly modular architecture, and it supports XLIFF standard. The suite consists of mainly three discreet modules: XLIFF Editor, translation memory engine (written by Fredrik Estreen) and filters to convert different document formats (such as XML, SGML, PO, RTF,StarOffice/OpenOffice)to/from XLIFF [13][14]. One of the major drawback of Transolution is the lack of GUI to combine the above mentioned components. Thus, in the

[1] TMX is described in section 5.2
[2] XLIFF is described in section 5.1

present configuration, one has to use Python command line to execute separate modules [12]. Moreover, the documentation is too technical making it difficult to understand for a layman though it is comprehensive.

XLIFF editor, the core application out of three modules, is the main working environment. XLIFF Editor is comparatively similar to the GUI of OmegaT and Wordfast and found to be attractive and user friendly. XLIFF presents a combined window in which translated segments contain the translation, untranslated segments the original source texts, and only the active segment both [14]. Screen capture of XLIFF editor is shown in Figure 2.

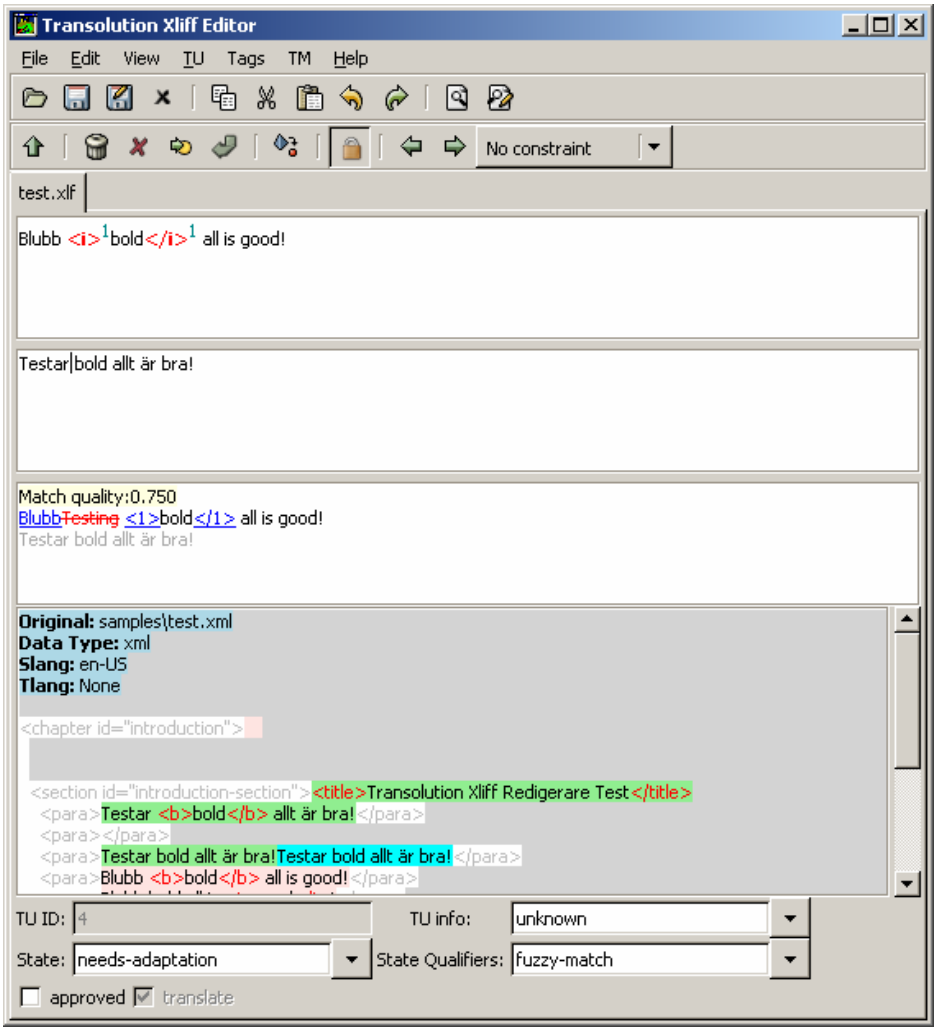

Figure 2: Transolution XLIFF Editor

At the time this report was prepared (2008), the *Frederik Corneliusson* has mentioned in the official website that Transolution development is on hold [13]. However Transolution is a working solution, Thus it can be expected that developers all around the world would improve it in the years to come.

4.3 Open Language Tools

A set of translation tools developed in Java for localization of the Sun's own documentations, later (in 2005) released for public as open-source tools [17][18]. These set of tools comprise of a full-featured XLIFF Translation Editor and a set of XLIFF file-filters for a number of documentation and software file formats including, HTML, Docbook SGML, JSP, XML OpenOffice: sxw, sxc, sxi, Open Document Format : odw, odc, odi, Plain text, PO (gettext), Msg/tmsg (catgets), Java .properties, Java ResourceBundle and Mozilla .DTD resource files.In the future, it is expected to develop a server-side translation memory tool (TM tool) to enhance the functionality.

These tools are made available under the Common Development and Distribution License (CDDL). Screen capture of Open Language Tools' XLIFF Editor is shown below in Figure 3.

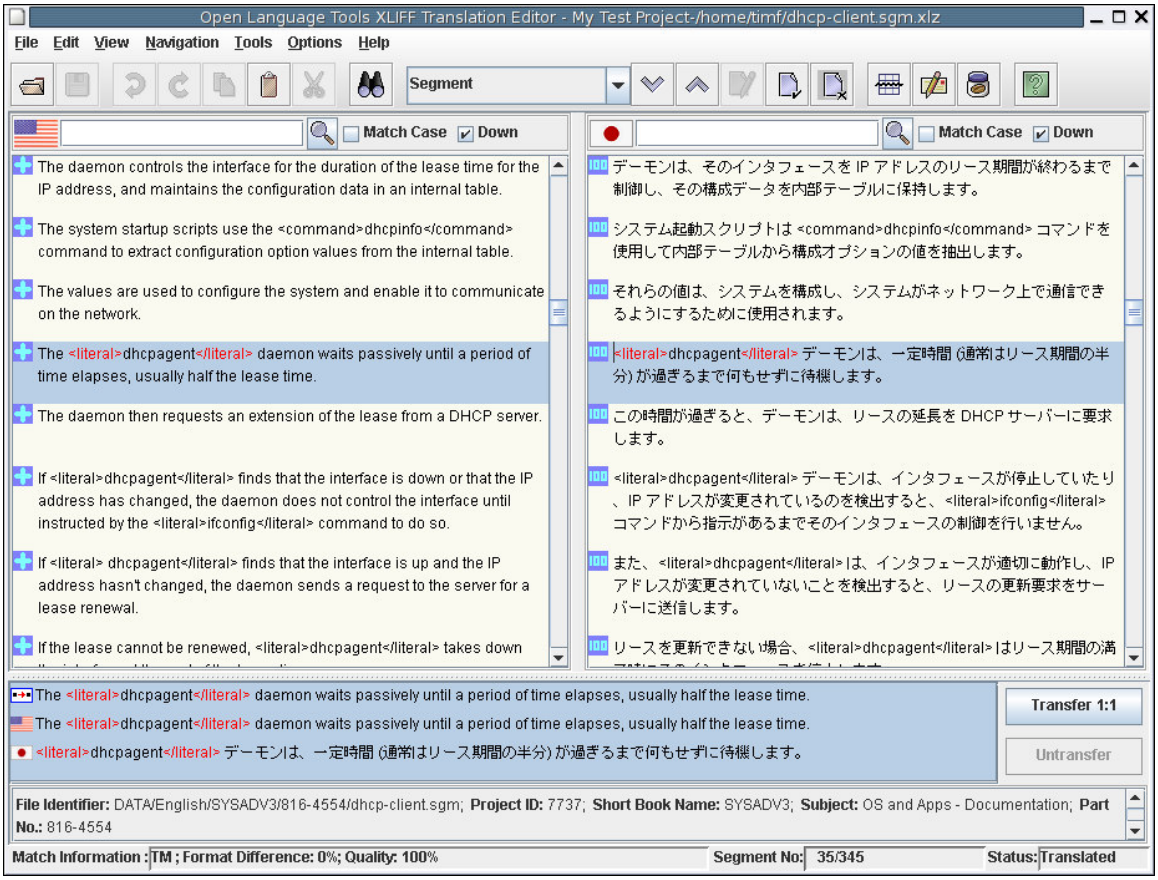

Figure 3. Open Language Tools' XLIFF Editor

4.4 ForeignDesk

ForeignDesk is an Open-Source Integrated Translation Environment for Windows. ForeignDesk was originally developed by Lionbridge[3] as a commercial software. Later, in 2001, it was made available as free and open-source application under BSD and IBM public license [15]. ForeignDesk suite is written in C#, C++ and Visual Basic [16]. ForeignDesk supports formats such as XML and it is TMX complient. It is expected that ForeignDesk would be further developed to support XLIFF standard and as an XLIFF editor.

[3] Company home page: http://www.lionbridge.com/

5. Open Localization and Language Technology Standards

Set of standards have been proposed by various organizations to implement interoperability among different organizations, individuals, TM (and localization)tools, and data exchange.

5.1 XML Localization Interchange File Format (XLIFF)

XML Localization Interchange File Format (XLIFF), is an emerging standard structured interchange file format for exchanging localization related data and metadata. It addresses various issues related to translation and localization process and it helps to optimize the localization or translation workflow.

XLIFF standard was developed in 2001, by a technical committee formed by representatives of group of companies, including: Oracle, Novell, IBM/Lotus, Sun, Alchemy Software, Berlitz, Moravia-IT, and ENLASO Corporation (formerly the RWS Group). In 2002, the XLIFF specification was formally published by the Organization for the Advancement of Structured Information Standards (OASIS) [19][20].

Purpose of XLIFF as described by the OASIS is to *"store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools"*[19]. By using this standard, data can be exchanged between different companies, organizations, individuals or tools such as TM systems. Various file formats such as plain text, MS Word, docbook, HTML,XML etc can be transformed into a XLIFF standard document, which enables translators to isolate the text to be translated from the text layout.

The XLIFF standard aims to [21]:

- Separate translatable text from layout and formatting data.
- Enable multiple tools to work on source strings
- Store metadata that is helpful in the translation/localization process.

The XLIFF standard is widely accepted as the de facto standard by all most all localisation service providers and supported by majority of localisation tools and CAT systems .The XLIFF Technical Committee at OASIS is continuingly improving the standard.

5.1.1 Extract-Localize-Merge Model

Figure 4 illustrates the translation process that uses XLIFF files as an intermediary format.

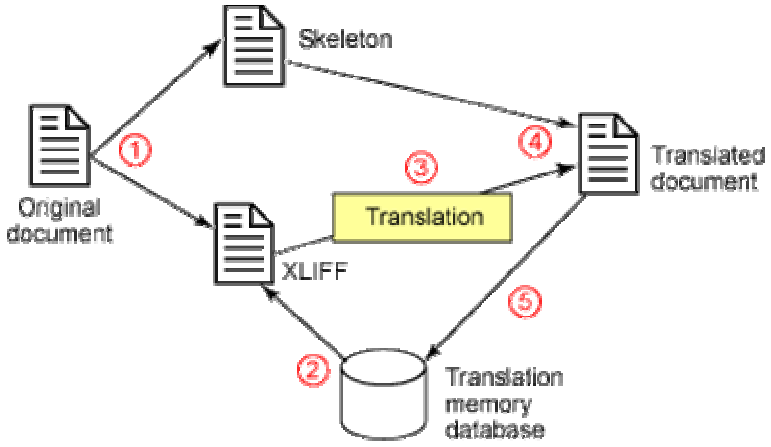

Figure 4. Translation Process using XLIFF as an intermediary format[4]

The first step involves conversion of source document in to XLIFF standard document. This process will separate translatable content from layout and formatting data. Formatting data along with placeholders for translated text will be written to a special file called "Skeleton". (This file can be embedded into the same XLIFF file too). In the second step, a TM tool can be used to pre-translate existing text to the XLIFF file. A professional translator can then review add missing translations or modify the translations in the third step. Upon the translation process, XLIFF is merged with the skeleton file to create translated XLIFF file, this is the fourth step. An optional fifth step would be to populate translation memory with recently translated data.

5.2 Translation Memory eXchange (TMX)

The Translation Memory eXchange format was defined by the Open Standards for Container/Content Allowing Re-use (OSCAR) , a special interest group of the Localisation Industry Standards Association (LISA).

TMX format defines a common standard to exchange translation memory data between TM systems.

The formal definition extracted from the TMX Web site [23] is given below:

> *"TMX (Translation Memory eXchange) is the vendor-neutral open XML standard for the exchange of Translation Memory (TM) data created by Computer Aided Translation (CAT) and localization tools. The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process."*

Until the TMX was defined, proprietary TM system developers refrained from making their TM databases accessible by the other similar TM systems. Main reasons for this including the loss of brand loyalty among users and increased competition among product developers [6]. Although, TM users felt a great need of a common standard to allow TM databases to be shared among

[4] Image retrieved from: http://www.ibm.com/developerworks/xml/library/x-localis2/

different users as well as different tools. As such, it became inevitable for TM system developers to built TMX support in their tools.

TM databases represent a significant effort and an investment that may even worth hundreds of millions of dollars. With the advent of TMX, now individuals and organizations can maintain and share these TM databases in a vendor-neutral and tool-independent format and reuse their content as much as possible. Reusing existing TM content will significantly reduce cost in translation projects.

TMX is currently at version 1.4b, which a certifiable standard and LISA expected to release TMX 2.0 in year 2008. Various tools (e.g. TMX compliancy validation tools, file format conversion tools, TMX editors) have been developed and published by various developers including LISA [23]. Today, all major CAT tools support TMX.

5.2.1 TMX Definitions & Levels

TMX is defined in two parts, and it offers two levels of implementation [23][24].

Two parts of definition:

- A specification of the format of the container (the higher-level elements that provide information about the file as a whole and about entries)
- A specification of a low-level meta-markup format for the content of a segment of translation-memory text. TMX offers two levels of implementation

Two levels of implementation:

- Level 1-Plain Text Only: Only translatable text is included in the TMX document, leaving formatting information aside. (Support for the container only.)
- Level 2-Content Markup: Text and markup information are included in the TMX document. (Support for both container and content.)

5.3 Term Base eXchange (TBX)

A glossary is a list of technical terms in a particular domain of knowledge. The glossaries are frequently used by the translators. Glossaries available in various file formats can be used with CAT tools to speed up the translation process. In order to eliminate the difficulties of having glossaries on multiple formats and standards, Term Base eXchange an XML-based standard for exchanging structured terminological data was developed by LISA. LISA has submitted TBX specification to the International Organization for Standardization (ISO), and it is expected this specification will be accepted by ISO (in ISO context: ISO DIS 30042) [25]. TBX format is suitable for creating, maintaining and sharing terminological databases, dictionaries and glossaries. Glossaries in TBX format can be readily used by major CAT tools.

5.4 Segmentation Rules eXchange (SRX)

Different languages have language-specific segmentation strategies. For some languages, word segmentation is a non-trivial task (e.g. as in CJK languages). Thus, for some languages it is almost impossible to segment a sentence perfectly. Though native speakers can identify the

segments, it is very difficult to implement automatic segmentation algorithms for most of these languages; this is especially due to the lack of linguistics rules for segmentation of words.

It is an obvious fact that TM systems should be able to work with multiple languages, and deal with multiple languages. Therefore, TM systems should be capable of segmenting text in several languages. In order to supports as many languages as possible, these tools should implement algorithms and rules to break text into different segments such as paragraphs, sentences, or words. In other words, the tools should be capable to handle various linguistic complexities in different languages. However, defining algorithms to deal with such complexities is a very difficult task. For an example, different tools identify and segment text in different ways. Having identified these issues, LISA defined a standard called Segmentation Rules eXchange (SRX) describing how translation and other language-processing tools segment text for processing [26]. This vender-neutral standard allows TM and other CAT tools to implement and the language-specific segmentation rules and algorithms. Furthermore, rules and segmentation information described using SRX can be shared among different CAT tools too. TM data can be deployed by coupling TMX data with SRX data, and combined data will definitely improve the functionality of TM systems.

6. Conclusion

The main objective of this study was to describe the state-of-the-art of TM technology and to propose methodology for development of an open source TM system.

Mandatory requirements of the proposed TM system:

1. It should have a standalone working environment
2. It should be platform Independent
3. It should support the most recent versions of all open language standards described above (XLIFF, TMX, TBX, SRX)
4. Multilingual: should be able to work with many languages as possible
5. The TM system should be localizable
6. It should have built-in-support for most widely used document formats (e.g. Microsoft Word Documents, OpenDocument Format, HTML, XML) (i.e. document converters and filters should be embedded to the system)
7. Should have a user-friendly efficient GUI
8. Should implement optimized search, match and retrieval techniques

Optional additional advance features include, the networking support where TM system can access multiple TM databases distributed in network environment. TM application can be further developed to run as a server. A TM's ability be used over a network will help an entire team to maintain consistency throughout the translation of many documents.

The findings of this study propose to further improve an existing open source TM system that deliver most of the above mentioned features. Improving an existing system will significantly cut down the development cost and effort. Towards this end, further development of OmegaT and Transolution application are considered.

Though OmegaT already has most of the expected features, some issues were noticed. OmegaT is written in Java. Therefore, inevitably, it has both advantages and drawbacks inherent to the Java platform. Moreover, some rendering issues still persist in Java Swing (or Abstact Windows Toolkit) for some languages (e.g. Sinhala), which is used to create OmegaT graphical user interface. Next best OS-TM system is Transolutions, which is entirely written in Python. Python is a free and open source scripting language. Python is becoming increasingly popular and embraced by the development community due to attractive features that it comprises of compares to many other computer languages. The use of Python is found to be significantly more productive in developments (especially in RAD) than using Java [27]. Another benefit is that programmers have the freedom to use vast number of libraries developed for Python contributed by the Open Source Python community. Python supports Unicode, and no issues were found related to rendering when creating interfaces. Moreover, Transolution is written in a modular manner. The modular arrangement will facilitate the further development process. Considering all these benefits, this study proposes to further improve Transolution OS-TM system to include all mandatory features described above.

References

1. *"Design and function of translation memory"*, EAGLES Evaluation of Natural Language Processing Systems FINAL REPORT, EAGLES DOCUMENT EAG-EWG-PR.2, Version of September 1995, Retrieved from: http://www.issco.unige.ch/ewg95/node152.html

2. *"Computer-assisted translation"*, Retrieved from: http://en.wikipedia.org/wiki/Computer-assisted_translation

3. Olivia Craciunescu, Constanza Gerding-Salas, Susan Stringer-O'Keeffe (2004), *"Machine Translation and Computer-Assisted Translation: a New Way of Translating?"*, Translation Journal, Volume 8, No. 3. July 2004, Retrieved from: http://www.accurapid.com/journal/29computers.htm

4. *"Translation tools and resources"*, Retrieved from :http://www.betranslated.com/translation-tools.html

5. *"Translation Memory Tools, Computer Aided Translation, Other Tools"*, Retrieved from: http://www.translatum.gr/dics/translation-memory.htm

6. Lynn E. Webb (1999), *"ADVANTAGES AND DISADVANTAGES OF TRANSLATION MEMORY: A COST/BENEFIT ANALYSIS"*, MA Thesis, Translation of German, Graduate Division, Monterey Institute of International Studies, Monterey, California.

7. Elina Lagoudaki (2006), *"Translation Memories Survey 2006, Translation Memory systems: Enlightening user's perspective"*, ASLIB International Conference 'Translating and the Computer 28', London, 15-16 November 2006.

8. Gow, Francie (2003), "*Metrics for Evaluating Translation Memory Software*." MA thesis. University of Ottawa, Canada.

9. Angelika Zerfaß, *"Comparing Basic Features of TM Tools"*, Language Technology, MultiLingual Computing & Technology Magazine, October/November 2002, MultiLingual Computing, Inc, 319 North First Avenue, Sandpoint, Idaho 83864 USA.

10. *"OpenTM- An Open Translation Memory Tool"*. Retrieved from: http://www.opentm.org/http://www.opentm.org/

11. *"Introducing OmegaT"*, Retrieved from: http://www.omegat.org/en/omegat.html

12. Dmitri Popov (2005), "*Translating With OmegaT",* Retrieved from:. http://www.linux.com/articles/42532

13. "*Transolution: An Open Source Translation Suite"*, Retrieved from: http://transolution.python-hosting.com/

14. Marc Prior, "*Transolution". Retrieved from:* http://www.marcprior.de/linux/tm/transolution.html

15. *"Lionbridge ForeignDesk® Is Rapidly Embraced by the Open-Source Community and End-Users"*, Retrieved from: http://www.lionbridge.com/lionbridge/en-US/company/news/lionbridge-foreigndesk-is-rapidly-embraced-by-the-open-source-community-and-end-users.htm

16. *"ForeignDesk"*, Retrieved from: http://sourceforge.net/projects/foreigndesk/

17. *"Open-Language-Tools"*, Retrieved from: https://open-language-tools.dev.java.net/

18. Marc Prior, *"Sun Open Language Tools"*. Retrieved from: http://www.marcprior.de/linux/tm/olt.html

19. *"XLIFF Version 1.2: OASIS Standard"*, Retrieved from: http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html

20. *"XML in localisation: A practical analysis: An overview of the most relevant XML standards used in the localisation industry"*, Retrieved from: http://www.ibm.com/developerworks/xml/library/x-localis/

21. *"XLIFF: An Aid To Localization"*, Retrieved from: http://developers.sun.com/dev/gadc/technicalpublications/articles/xliff.html

22. *"XML in localisation: Use XLIFF to translate documents"*, Retrieved from: http://www.ibm.com/developerworks/xml/library/x-localis2/

23. *"Translation Memory eXchange (TMX)"*, Retrieved from: http://www.lisa.org/Translation-Memory-e.34.0.html

24. *"XML in localisation: Reuse translations with TM and TMX: Reduce translation time and effort with the aid of XML standards"*, Retrieved from: http://www.ibm.com/developerworks/library/x-localis3/

25. *"Term Base eXchange (TBX)"*, Retrieved from: http://www.lisa.org/Term-Base-eXchange.32.0.html

26. *"Segmentation Rules eXchange (SRX)"*, Retrieved from: http://www.lisa.org/Segmentation-Rules-e.40.0.html

27. *"Python & Java A Side-by-Side Comparison"*, Retrieved from: http://www.ferg.org/projects/python_java_side-by-side.html

28. *"Dr. Tom's Independent Software Reviews: Software Localization Tools Comparison Chart and General Conclusions"*, Retrieved from: http://www.localizationworks.com/DRTOM/Conclusions.html