Špela Vintar

Department of Translation

University of Ljubljana

Real-World Projects in Localization Training

1. Introduction

Courses on eContent Translation and Software Localization have become an important and an increasing popular part of translator training curricula all over the world. At several translation institutes localization is considered a field of study in its own right, with BA and MA programmes specializing in Language Services and Localization (Damiani 2002; Altanero 2003). However, the majority of European institutes, including Ljubljana, provide Software Localization as a specialized module within a general translation curriculum at the MA level.

There are quite a number of skills to be conveyed within a course on Localization. Provided the students have already been introduced to CAT tools and terminology management within courses taken previously, a Localization course could include any or all of the following (Lohrer 2006):

- a revision of eContent-related terminology and a review of terminology resources,
- localization basics: processes, tools, people involved etc.,
- a thorough introduction to file formats encountered in the translation of executables, language files and documentation, including web sites,
- an introduction to XML,
- an introduction to desktop publishing (DTP),
- localization tools,
- character encoding issues and locales,
- project management (PM) and related tools,
- the ethics of localization and culture-related issues,
- legal aspects of localization.

Since the localization process in practice involves such a large variety of tasks performed by a number of specialized people, one problem that arises in an educational setting is providing an adequate instructor for the course, one who ideally masters all the above mentioned skills and is at the same time a good trainer from a didactic point of view.

An important way to learn things like project management and team localization, but also to put technical and linguistic skills into practice, is project work. Within a real localization project, where a software product, a web site or a bulk of product documentation is to be localized within a limited time period and for a known client, the students need to prove themselves in an entirely different way than in traditional translation classes. Personal qualities like reliability, co-operativeness, leadership skills and ability to work under pressure are as important as translation skills.

In the remainder of this paper three projects are discussed that were performed within a Localization course at the Department of Translation,[1] University of Ljubljana, in three consecutive years. The aim is to show that a number of factors need to be considered in planning such projects and that, ideally, student projects are managed by students not teachers.

2. Materials and resources

Each localization project involves the use of certain tools that facilitate the process of adapting the product to the local market. These include tools for the extraction of translatable elements from files, DTP tools, project management tools, terminology management and/or terminology extraction software, translation memory tools, editors for different file formats etc. Some or all of these features

[1] http://www.prevajalstvo.net

may be included in a piece of localization software, which may in addition offer a WYSIWYG mode so that the strings to be localized are also visible in context.

For the projects we describe below, one or several of the following tools were used:

- TRADOS[2] TagEditor and Translator's Workbench, version 6.5 (licensed)
- MultiTerm iX (licensed)
- poEdit[3] (distributed under MIT License)
- Language Manager[4] (free)
- Help & Manual[5] (demo version)

Obtaining the software for localization training is often related to financial issues, but still presents a minor problem compared to the task of obtaining suitable projects, in other words clients willing to release their product into the hands of localization students. The gap between the concerns of a potential client and the aims of a localization teacher seems almost too broad to be bridged. The teacher ideally wishes for a project with the following characteristics:

- It should include an entire software product or at least a manageable software component in its entirety.
- It should be of an appropriate size and format to be divided among students and then reassembled.
- The software itself should be available to students during the localization process.
- It should not be too specialized, otherwise the students will be unable to understand the program functions and will produce inadequate translations.
- It should run on a Windows platform.
- It should not already exist in the target language.

The software vendor on the other hand sees extremely little benefit in co-operation with a translation institute. In most cases, the software components to be made

---

[2] http://www.trados.com
[3] http://www.poedit.net/
[4] http://devtools.korzh.com/downloads/
[5] http://www.ec-software.com/

available for translation are under copyright protection, and even if non-disclosure agreements are signed both parties may still feel uncomfortable because a large group of students is involved. Furthermore, translations produced by students may not meet the required quality standards. Even if the localization project is well-prepared and thoroughly revised, no localization trainer can guarantee a flawless result.

Localization trainers have been well aware of this problem for some time. A milestone in providing free training materials was the eCoLoRe project,[6] a predecessor to eCoLoTrain. eCoLoRe assembled and made public an astounding set of multilingual materials in various formats, translation memories and guidelines covering all main aspects of localization. Fortunately, Slovene is one of the languages included in the eCoLoRe toolkit, so that the materials have proved invaluable at several teaching occasions. Still, the materials there are not suitable for a localization project where the students are to experience all the pressures and complexities of a real-world project.

In the course of the past six years since the beginning of localization training at the Department of Translation Ljubljana, several language service providers and software vendors have been approached with a request for co-operation with our institute, both in Slovenia and abroad. In the majority of cases we received no reply at all, and where there was a response, it was far from enthusiastic. For the above reasons, the only feasible solution was to turn to the open source community, where software internationalization is based primarily on voluntary work performed by non-professional translators. The Slovenian Linux community, organised into a non-profit association LUGOS,[7] were more than willing to accept our offer of providing free student translations and in return offered technical support and advice. From this co-operation several student projects were born, of

[6] http://ecolore.leeds.ac.uk/
[7] http://www.lugos.si

which two are described in more detail below, the KDE and the OpenOffice projects.

The third and most recent project presented here involved neither an open source product nor a highly commercial one. Wordsmith is a tool best known in linguistic circles, and it was indeed a happy coincidence that its author, Mike Scott, immediately agreed to Wordsmith Tools being localized into Slovene. Below we briefly present all three projects and evaluate their success in view of the educational goals.

3. Projects

In all projects we describe below, the class of 20-25 students was divided into several groups of 4-5, each consisting of a co-ordinator, terminologist and 3-4 localizers. The entire project workload was divided equally among the groups, whereby further division of labour among individual group members was left to the group co-ordinators. After the introductory plenary session during which the goals of the project were explained and the technical requirements given, there were no traditional classes for the remaining duration of the project. Instead, consultation meetings were organised at regular intervals with individual groups, with all the terminologists and all group leaders respectively. The latter were required to report on the progress and problems encountered during the project as well as submit a written final report evaluating all aspects of team work in their group.

In all projects, one deadline was given for submission of the first version, and another deadline for the submission of the final revised version.

The role of terminologists was to collect terminology from all available sources, offer support to their team members, consult with fellow terminologists and ensure

term consistency throughout the project by creating, updating and distributing the project term bank.

In the remainder of this section each project is described from a number of aspects. The **size of project** and **group size** parameters give the reader an idea about the workload assigned to each individual student, the **type of project** and its **description** provide background information about the project content, the **level** is a subjective assessment of the difficulty and complexity of the task presented to the students and ranges from medium to complex, the **platform**, **file types** and **tools used** give information on various technical aspects of the projects, under **background materials** we list the resources available to students, such as previous translations of similar material. Another aspect is the amount of **project management skills** that had been conveyed to students prior to project beginning (if any), the nature and amount of **client support** (if any), and whether **testing** (Esselink 2000: 178-180) was included as part of the project.

3.1 KDE

Size of project	ca. 125.000 words
Group size	24
Type	documentation: online help
Description	The KDE desktop environment is a large set of utilities and tools for unix workstations. It includes system utilities, programming tools, an entire office application KOffice, networking solutions such as KMail, games etc. It is distributed under GNU GPL and has so far been completely or partially localized into over 100 languages, including Slovene.
Level	complex
Platform	Unix/Linux
File types	.po
Tools used	Trados TagEditor, poEdit, Trados MultiTerm
Background materials	A database of English-Slovene strings from localized KDE applications was provided by the client.
Project management skills conveyed	none
Client support	medium
Testing	no

The project was initiated on the basis of a history of good relations between our translation department and the Linux users group of Slovenia, Lugos. Within Lugos, a section of translators co-ordinated by Andrej Vernekar was at the time (January 2005) working on the localization of KDE docs into Slovene, while the majority of the GUIs had already been localised. Our students were thus needed mainly to help out with the documentation, of which selected packages were suggested to us as priority. The project finally contained 39 files pertaining to packages KMail, KAddressbook, Knotes, Kolourpaint, Korganizer and Kopete.

The source files were available from the kde.org site in .po format, which is a standard file format for language files in the open source community. In order to be able to translate them with TagEditor, it was necessary to convert the .po files into XML, which was done using custom-made Perl scripts. After translation the target XML files were converted back into PO.

Since KDE belongs to the Linux world exclusively, and since none of the students involved was at the time a Linux user, it was impossible to make sure that each translator would have access to the application she was translating. Linux with a Slovene version of KDE was installed on one computer in the PC-lab, and the chief terminologist installed it on his laptop. As a result, hardly any translator even tried to look up the translated strings in the GUI, and those who did often complained they could not locate the functions or menu options within the program, or they even could not locate the application itself!

Although the students all had a MultiTerm database containing all the strings from the GUIs at their disposal, not knowing their context nor the functionalities of the translated applications rendered them incapable of producing adequate translations of the help files.

During revision, many translation errors were identified and corrected by the terminologists, however many still remained undiscovered. Due to time constraints

and the effort involved in back-conversion of files into .po format, no testing was included in the project.

The overall quality of the translations produced was poor. The main source of errors was incomprehension, resulting in blind word-for-word translations of the source sentences, followed by a general unfamiliarity with Linux terminology, thus producing erroneous translations of terms like *shell, root, console, distribution* etc. Apart from linguistic and translation-specific errors, several files were submitted with technical flaws. In most cases, the target XML file was not well-formed due to missed tags or corrupted entities. A large portion of the time allocated to revision was spent correcting technical flaws, thus neglecting the linguistic quality.

Despite those difficulties, the group dynamics and the students' self-management worked surprisingly well. Except for one person, everyone handed in their translations by the deadline, and due to the chief terminologist's resourcefulness, most technical problems were solved by the students themselves without requiring intervention.

At the end of the project, the students were positive about it by pointing out what they had learned or gained:

- effective handling of XML files using TagEditor and web browsers,
- a great deal of IT-related terminology,
- a glimpse of the Linux environment,
- the characteristics and style of online help files,
- the experience of working in a team.

3.2 OpenOffice

Size of project	22.500 strings, ca. 250.000 words
Group size	25
Type	documentation: online help
Description	OpenOffice is an application suite similar to MS Office, including a word processor, a spreadsheet solution, a program for presentations etc. It is open-source and free,

	completely or partially localized into 96 languages, including Slovene.[8]
Level	medium
Platform	Windows, Unix/Linux
File types	.xml (DocBook)
Tools used	Trados TagEditor, Trados MultiTerm
Background materials	The already localized GUI was available along with a database of translated strings.
Project management skills conveyed	none
Client support	yes
Testing	no

For this project we were approached by Robert Ludvik, the co-ordinator of the OpenOffice Slovenia project, who suggested we help by translating portions of online help for OpenOffice 2.1, of which the interface had already been localised into Slovene. In technical respects, the project was similar to KDE. The bulk of our project comprised the entire documentation for the Calc application, the OO component similar to MS Excel, plus some shared files containing strings pertaining to no specific OO application.

The main difference was the fact that OpenOffice runs both on Windows and Linux, so that the students were able to install the already translated application on their machines and refer to it throughout the translation process. Apart from this, a database of the translated strings from the GUI was made available, which we converted into MultiTerm format and distributed among the group.

Although most of the students had access to the application in the target language, many sections of Calc documentation proved difficult to translate. The program contains numerous mathematical functions that need to be understood in order to be described in a readable and clear fashion. Also, mathematical terminology was

[8] http://sl.openoffice.org/

not adequately handled in many cases, as the example below shows. The term *conjugated complex complement* should have been translated as *konjugirani kompleksni komplement.*

> Original: The result is the conjugated complex complement to a complex number.
>
> Translation: Rezultat je povezan kompleksni dodatek h kompleksnemu številu.

Another common source of errors were locales such as the decimal point that should have been translated into the decimal comma in Slovene:

> Original: The constant e has a value of approximately 2.71828182845904.
>
> Translation: Konstanta e ima približno vrednost 2.71828182845904.

The conclusion of the project was its semi-public presentation on the premises of the client's company, where each student group described their workload, the problems encountered and the experience gained. The client even presented each student with a USB memory stick as an acknowledgment of their efforts.

3.3 Wordsmith Tools

Size of project	4925 strings of the GUI, ca. 30.000 words of the user's manual
Group size	20
Type	GUI, documentation: online user's manual
Description	Wordsmith is a set of tools for processing and exploiting corpora, mainly for the purpose of linguistic analysis. It includes a concordancing tool, a tool for building word lists and computing text statistics, a tool for keyword extraction and a range of file manipulation utilities. It is developed and maintained by Mike Scott and has so far been localized into German, French and Slovene.
Level	medium
Platform	Windows
File types	.lng, .xml
Tools used	Language Manager, Help&Manual, Trados TagEditor, Trados MultiTerm
Background materials	none
Project management skills conveyed	yes, basics
Client support	medium
Testing	yes

Since Wordsmith had previously been localized into German and French, Mr Scott had some suggestions concerning the methods and tools to be used. Language Manager, the free localization component of Localizer, was suggested for the translation of .lng files, ie. strings from the GUI, and the demo version of Help&Manual was used for decompilation of .chm files into .xml, which were then translated using TagEditor.

In contrast to both projects described above, here the students were made familiar with the basics of Project Management, particularly the roles and responsibilities of different team members, the importance of efficient folder structure, time and resource management etc (Matis 2005). Although these contents were presented to the students in a single 90-minute session, I am convinced that this know-how was

extremely beneficial for the project's success. The PM course was taught using eCoLoTrain materials.[9]

Another key difference to previous projects was parallel translation. The class of 20 students was divided into 4 localization groups consisting of 4 persons each, a group of revisers consisting of 3, and a chief project manager/terminologist. Then, the entire workload was assigned to 2 plus 2 localization groups, meaning that all work was done twice. The task of revisors was then to compare both translations of each project file, consolidating the differences and selecting the preferred translation. This division of labour proved efficient and successful for several reasons:

- Parallel translation motivates students to take their jobs more seriously, as they know each of their solutions will be compared to another person's translation.
- Parallel translation makes it possible to identify problem strings automatically.
- If each file is to be translated twice, it means that in the end there will exist at least one translation of all files, in case certain individuals choose to depart from the group or fail to submit their translations.

For the translation and revision of the GUI, which comprised approx. 5000 strings, the students were given a deadline of 4 weeks, plus another 2 weeks for testing. Wordsmith runs on Windows and was available to students in all existing language versions, English, French and German. As no other similar tool had ever been translated into Slovene, terminology research required substantial efforts.

When the author of the software made available a first Slovene Wordsmith release for testing purposes, the students were thrilled. While in real-life localization projects the first glimpse of a newly translated product might be a dreadful moment in which all kinds of hidden errors come afloat, here the joy of seeing the

[9] http://ecolotrain.uni-saarland.de/index.php?id=1924&L=1

result of their work was touching. Nevertheless, the testing of the product was taken very seriously and many errors, mainly due to exceeded string length, were corrected. Figure 1 shows an example of such an error, where the original string *.zip files too* was translated into *tudi datoteke .zip*. Such errors may result in severe disfunctionalities of the localized product.

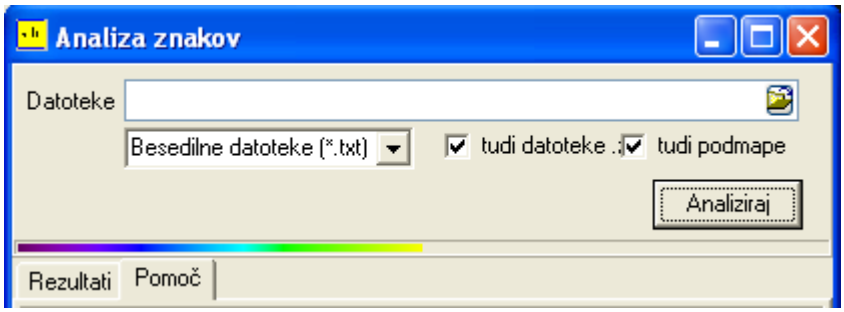

Figure 1: Exceeded string length in Wordsmith Tools

The quality of the final version was good both from the stylistic and the technical point of view. Throughout the project, all groups respected the deadlines and showed exceptional zeal and enthusiasm. In cases where certain group members did not produce satisfactory translations, this was efficiently handled within the group either through peer revision or redistribution of labour.

At the end of the project the students' feedback was positive. Complaints were expressed regarding the localization tool used (Language Manager), especially because the tool does present the strings in context, nor does it offer any help in locating errors.

4. Conclusion

In summary, a comparison of the three projects described above shows that certain scenarios work better then others. The characteristics of an ideal localization project in an educational setting can be summarized as follows:

- Size of project (for student groups around 20 persons): Up to 7,000 strings GUI and up to 250 pages documentation

- Type: Complete software product, with GUI and docs
- Level: Medium
- Platform: Windows
- Background materials / Previous versions: Not absolutely necessary
- Testing as part of the project: Absolutely necessary
- Project Management skills prior to project beginning: Not absolutely necessary but extremely helpful
- Client Support: Same as above.

Whether such projects can in fact be obtained for educational use remains a problematic issue. Clearly the source of suitable software products remains in the area of specialized utilities offered as open source or at symbolic fees, while large software vendors will probably continue to avoid such cooperation. Since such projects contribute a valuable and fun experience to future localization experts, their pursuit remains worth the effort.

References

Altanero, T. (2003) The localization job market in academe. Localization and Translator Training, an online conference 20-29 October 2003. [URL: http://isg.urv.es/seminars/2003_localization_online/altanero.html]

Damiani, R. (2002): Localization Education in Montréal. Globalization Insider (LISA), 10/2002. [URL: www.lisa.org/globalizationinsider/2002/10/localization_ed.html]

Esselink, B. (2000): A Practical Guide for Localization (2nd Edition). John Benjamins. ISBN 1588110060.

Lohrer, M. (2006): Globale Software. HighTech 12 – 2006. p. 138-143. www.donetpro.de [URL: www.passolo.de/files/dotnetpro1206.pdf], May 2007.

Matis, N. (2005): La gestion de projets de traduction et sa place dans la formation de traducteurs, *Équivalences*, number 32/1 2005 - "La traduction à l'heure de la localisation" HEB, Haute École de Bruxelles, 47-62.