

<Article>

## Development and Evaluation of Localization Training Software

Ryutaro Nishino and Kayoko Nohara

Tokyo Institute of Technology

While software localization and internationalization are inseparable processes, they are often done in different locations by different professionals: translators and developers. This division of labor causes problems such as translators not having a chance to check the context when they translate software user interfaces (UIs) or not having the knowledge of internationalization. This paper introduces software that is designed to allow translators to translate UIs and check the context and result instantly. The software can also be used to teach different elements in different locales (e.g., calendar, date, number formats) and how to use CAT tools. Although the result of a small questionnaire study shows that the software has a certain level of educational effectiveness, further research is needed to determine its effectiveness.

### 1. Introduction

#### 1.1. Background

Software localization and internationalization are a series of processes that cannot be separated. Globalization and Localization Association (GALA), an industry organization, defines "localization" as follows (Globalization and Localization Association, n.d.):

Describes the process of adapting a product to a specific international language or culture so that it seems natural to that particular region, which includes translation, but goes much farther.

Localization includes work related to translation. GALA also defines "internationalization" as follows:

The process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for redesign.

This process is related to software engineering. Engineers or developers internationalize a software product, and translators localize it. In this way, localization and internationalization are inseparable processes. The software company Apple Inc. (n.d.) also says "Internationalization and

localization are complementary activities." However, in the current internationalization and localization process, developers and translators often work in different locations. Although software products were internationalized and localized in a software company in 1980s (Esselink, 2000, p. 6), the localization process was separated and outsourced to localization companies thereafter.

There are some problems arising from this division of labor. First, translators are often forced to translate user interface (UI) strings without seeing the actual UI (Ressin et al., 2011), resulting in their assumption that the localization quality is low. When translators cannot see the context of strings, they have to use their limited knowledge to imagine it. Second, translators often do not know about internationalization, which is complementary to localization. This may cause translators to mistakenly delete placeholders in a string, for instance.

What can the localization education do to cope with such problems? With the first problem, the experience of translating UI strings and then checking them on the actual software UI can be effective. After experiencing such training, translators should have a better imagination about contexts. With the second problem, teaching the knowledge about internationalization to translators should be effective. With such knowledge, the quality of translation skills along with project management skills can be improved because translators or project managers know about the entire process of internationalization and localization.

Similar localization education projects have been held before. Such an example is the 10th Annual LRC Localisation Summer School. Participants developed mobile applications for Android or Windows Phone and localized the strings (Localisation Research Centre, 2012). There are educators who adopt the approach of translating and checking strings on actual UIs.

## 1.2. Purpose

Software that allows translators to translate UI strings and shows them instantly on the UI should be useful in a localization education, whether in a classroom or in a self-training occasion. It is also desirable that such an application can be used to teach localization and internationalization. Thus, the authors developed localization training software for the purpose of training translator's imagination about contexts and improving the skills and knowledge about localization and internationalization.

Professional translators usually translate UI strings without displaying their translations on the UI while translating in an actual work environment. Therefore, not only students who are learning about software localization but also professional translators will benefit from using the training software.

This paper will introduce and describe the software developed for the above-mentioned purpose. To be more specific, (1) the requirements discussed in the design process, (2) the software functionality, (3) knowledge and skills translators can learn, (4) how the software is implemented,

and (5) the result of small evaluation study. The authors also explain the plan for a further research regarding the educational effectiveness and the software improvement plan.

## **2. Requirements for localization training software**

Before designing and implementing the localization training software, requirements were set for the training purpose as follows:

### a) Translators can check translation results instantly

The software should allow translators to switch the display language easily and check the context and validity of translated strings.

### b) Translators can learn about localization

Computer-aided translation tools (CATs) are indispensable in the modern localization. The training software is a translation target rather than a CAT tool, and translators can learn about how to use a CAT tool by translating the software with a CAT tool. For this functionality, the software should include a file format that can be translated with a CAT tool. It is also desirable that the software use different document types (such as UI or help) and file formats (such as HTML or JSON). Regarding the localization curriculum, O'Hagan (2006) says that a general overview of the localization industry, tools such as TMs and terminology management systems, basic computing knowledge (such as different file formats, file management, tags and character sets) should be taught in translator training programs.

### c) Translators can learn about internationalization

Because localization and internationalization are a series of processes, the software should help translators to learn about internationalization at the same time. Topics related to internationalization include: character encoding, the separation of source code and language resource, elements that differ in different locales (e.g., calendar, date format, currency, number format, noun plurals, direction of writing). With regard to software internationalization, Esselink (2000, p. 31) points out that the most important issues are character encoding, location of translatables (e.g., user interface text), user interface design, and regional standards. The last regional standards include elements such as calendar or number formats.

### d) Non-engineers can use easily

The software should be able to be manipulated using basic tools such as web browsers or text editors that are normally installed in personal computers, without the need of additionally installing a development environment that programmers use such as Eclipse. Setting a special development environment takes time, resulting in less time for a training session.

Based on the requirements explained above, we developed localization training software.

### 3. Localization training software: Expense Recorder

#### 3.1. Functionality from software users' point of view

In a localization training course, translators often translate static texts extracted from software. Because the authors think that a localization training course should use such real-life software that functions properly when a button is pressed or a menu item is selected. As an example of real-life software, the authors designed an application that allows software users to record expenses in an office. When someone spends money for something, they can enter information such as date or amount of money, and the software calculates the total amount of expense by category. Being a web-based application, it runs on a web browser. The entire screen is shown in Figure 1.

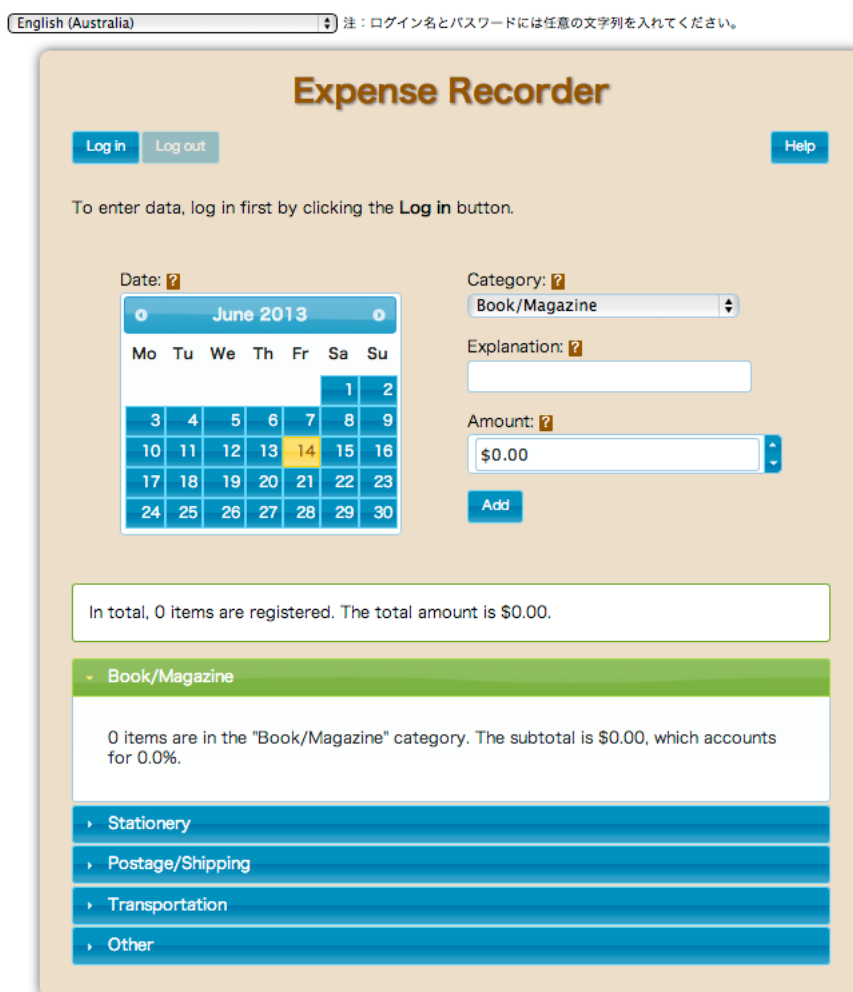


Figure 1: Entire screen of Expense Recorder

Functions including storing data and registering users have not been added because such functions are not directly related to localization.

### 3.2. Functionality from translators' point of view

The software has the functionality to switch UI languages. This allows translators to check if their translated strings are suitable in a context. From the perspective of localization education, this is the main functionality.

The software loads the language resource file of a default locale, and when a translator switches the locale, it loads the language resource file of the locale and displays the UI using the file (Figure 2). After translating the resource files, translators can check translated strings on the screen by switching the locale.

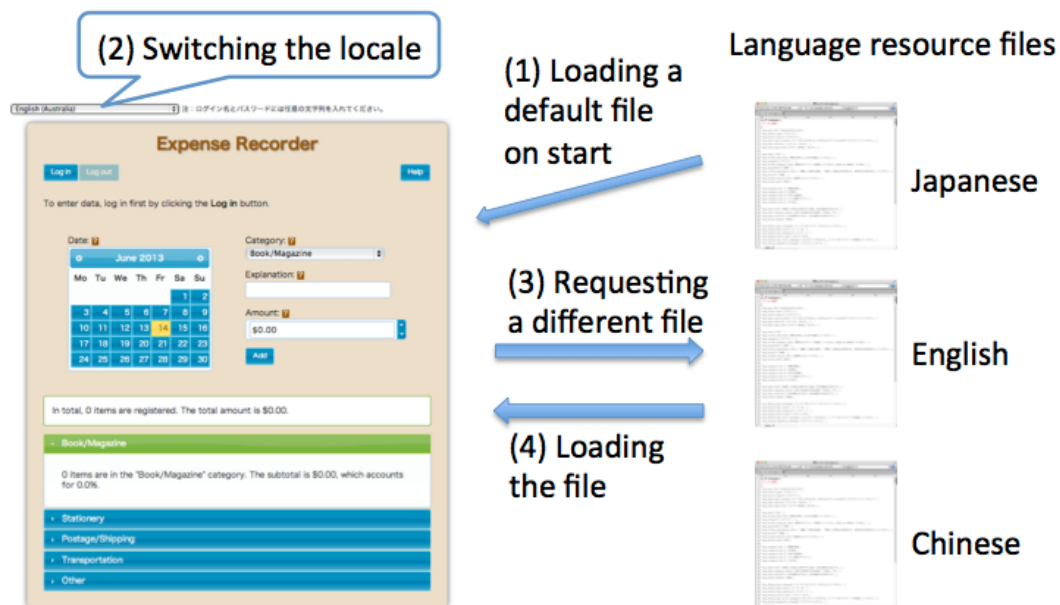


Figure 2: How switching a locale works

### 3.3. Knowledge and skills translators can learn

By using Expense Recorder as training materials, translators can learn about the following:

#### a) How to translate UIs

Translators will be able to improve their UI translation skills by translating UI strings such as button, menus, or messages and then checking the results on the screen.

One of the difficult tasks in UI translation is how to translate placeholders in strings. Unlike book translation, software generates data dynamically. Generated data is put in a

placeholder, and the order of placeholders is sometimes changed depending on the structure of a language. Thus, translators need to understand how placeholders work.

```
"msg_label_subtotal" : "The subtotal is %1$s, which accounts for %2$s%. ",  
"msg_button_delete" : "Delete",
```

Figure 3: Placeholders in a string

Translators can also learn that there are languages with multiple noun plural forms and how to deal with them. For example, while Japanese has one (no plural form) type and English has two (singular and plural) types, some languages have six different types of noun plural form.

```
"msg_label_category_items_zero" : "",  
"msg_label_category_items_one" : "%1$s item is in the \"%2$s\" category. ",  
"msg_label_category_items_two" : "",  
"msg_label_category_items_few" : "",  
"msg_label_category_items_many" : "",  
"msg_label_category_items_other" : "%1$s items are in the \"%2$s\" category. ",
```

Figure 4: Different translations for different plural rules

#### b) How to translate help documents

Translators can display help documents in HTML files by pressing the help button on the software. Help and manual translations are a major part of localization work, and translators can learn how to translate such documents.

#### c) How to use a CAT tool

In translating the software UI and help documents, translators can practice using a CAT tool. The HTML files of the help can be translated with a common CAT tool (e.g., Trados). The JavaScript file of UI strings can also be translated with some tools, such as OmegaT with its JSON filter.

#### d) Formats in different locations

The software has already been internationalized, and thus calendar, date, currency, and number formats are displayed in accordance with locales without the need of localization. Figure 5 shows the amount of money in French locale, where a comma is used for a decimal separator, a space is for thousand separators, and the euro symbol is for a currency mark. By switching locales, translators can see different formats even if UI strings are not translated yet. The software as of version 0.3 has been internationalized in 15 different locales.



Figure 5: Number and currency format in the fr-Fr locale

#### e) Scope of internationalization

One of the internationalization tasks is to externalize language resources from source code. Another task is to develop a mechanism to display information in different formats (e.g., calendar, currency) according to locales. Figure 6 shows a screen that is originally developed in English and internationalized, but not yet localized. When the screen language is switched to Japanese, the calendar and the currency format are displayed in Japanese because the software has been internationalized. However, since localization is not finished yet, other strings are displayed in the original English. Translators need to localize only these strings. When learners see this screen, they can understand that the part that developers deal with is internationalization and the part that translators deal with is localization.



Figure 6: Internationalized screen (but not yet localized)

## 4. Implementation

This section explains considerations for implementing the software. Those who are interested in technical aspects of the software or those who may develop similar training software in the future can refer to the technical information in this section.

#### a) Using Unicode

The software uses UTF-8 in HTML and JavaScript for the purpose of supporting different characters.

b) Using a standard locale format to support various locales

The software uses the combination of ISO 639-1 (language code) and ISO 3166-2 (country/region code), such as "en-US" or "ja-JP." The software supports 15 locales as of version 0.3 and can add supported locales.

c) Different strings can be loaded depending on the noun plural form

The software supports six plural types that The Unicode Consortium (n.d.) defines (zero, one, two, few, many, other).

d) Internationalizing components as far as possible

Calendar, date, number, and currency formats have been internationalized in the software. To be more specific, jQuery (<http://jquery.com/>), jQuery UI (<http://jqueryui.com/>) and related plugins are used.

e) Externalizing language resources as separate files

The software uses JavaScript files (.js) rather than XML or JSON formats to store language resources. This is because some web browsers (e.g., Google Chrome) prohibit reading files from a local hard disk. To overcome this issue, JavaScript files are used. This implementation method may not be preferable since translators can directly edit variables. To externalize resources, a library called jQuery-i18n (<https://github.com/recurser/jquery-i18n>) is used.

f) Folder structure

The structure of folders is shown below. The language resource files are separated from the main part of the software (ExpenseRecorder.html) and libraries (libs). The language resource files for UI are located in the "language\_ui" folder and files for help are in the "language\_help" folder.

ExpenseRecorder.html	<i>Main web application file</i>
libs	<i>Folder containing source files</i>
language_ui	<i>Folder containing UI language files</i>
en-US_messages.js	
ja-JP_messages.js	
zh-CN_messages.js	
...	
language_help	<i>Folder containing help language files</i>
en-US_help.html	
ja-JP_help.html	
zh-CN_help.html	



...

## 5. Evaluation

We conducted a small questionnaire study to determine if the localization training software has educational effectiveness. This study is also regarded to be a preliminary study to prepare for a full-scale research in the future. Participants are those who attended and used the software in a seminar held by Tokyo Institute of Technology, and they are 14 persons in total. Their occupations are students, engineers, and translators. After one hour of lecture on what localization and internationalization are and how to use the software, we had a hands-on style translation training using the software. They used text editors only (No CAT tools). At the end of the seminar, they responded to the questionnaire (with a consent to our publishing results in a paper). The questions and results are as follows:

### A. What is your occupation?

1. Undergraduate or graduate student	9	64.3%
2. Professional software developer	3	21.4%
3. Professional translator	2	14.3%
4. Other	0	0.0%
	14	100.0%

Table 1: Answers to the question A

### B. Checking the translated strings on the software user interface helps you to translate better because you can grasp the context.

1. Strongly agree	12	85.7%
2. Mildly agree	1	7.1%
3. Neither agree nor disagree	1	7.1%
4. Mildly disagree	0	0.0%
5. Strongly disagree	0	0.0%
	14	100.0%

Table 2: Answers to the question B

C. You learned that software translation has unique difficulties such as handling placeholders.

1. Strongly agree	9	64.3%
2. Mildly agree	4	28.6%
3. Neither agree nor disagree	1	7.1%
4. Mildly disagree	0	0.0%
5. Strongly disagree	0	0.0%
	14	100.0%

Table 3: Answers to the question C

D. You learned that there are different formats by locale such as calendar, date, currency, and number.

1. Strongly agree	12	85.7%
2. Mildly agree	2	14.3%
3. Neither agree nor disagree	0	0.0%
4. Mildly disagree	0	0.0%
5. Strongly disagree	0	0.0%
	14	100.0%

Table 4: Answers to the question D

E. You learned what localization is and what kind of work is included in it.

1. Strongly agree	10	71.4%
2. Mildly agree	3	21.4%
3. Neither agree nor disagree	1	7.1%
4. Mildly disagree	0	0.0%
5. Strongly disagree	0	0.0%
	14	100.0%

Table 5: Answers to the question E

F. You learned what internationalization is and what kind of work is included in it.

1. Strongly agree	8	57.1%
2. Mildly agree	5	35.7%
3. Neither agree nor disagree	1	7.1%
4. Mildly disagree	0	0.0%
5. Strongly disagree	0	0.0%
	14	100.0%

Table 6: Answers to the question F

The answers are generally positive. The reason may be because many of them learned about localization and internationalization for the first time.

When we look at the answers to the question B, the ability to check the translated strings in the UI clearly enhances the subjective quality of translation. Because professional translators do not have much chance to check the software UIs when they translate strings, this experience should improve the imagination about contexts.

From the answers to the question C, participants learned about a placeholder by actually manipulating it while translating the strings.

The questions D through F are about the knowledge of localization and internationalization. The answers show that the software helped participants to understand it. However, because we gave a lecture on locales, localization, and internationalization beforehand, the lecture might have affected the result more than the software itself has. Maybe we should think that the educational effectiveness comes from the combination of lecture and hands-on training, not the training alone.

We also asked participants to freely give their opinions about the software during the discussion session after the seminar. A professional translator, who is also an instructor of translation courses, admitted the positive educational effect and the potential as teaching material.

## 6. Future work

We think that further work is required in the following areas:

a) Research on the educational effectiveness

Further research on the educational effectiveness of the software is necessary. The small questionnaire study we did is rather a preliminary research, and we have to elaborate questions and research methods for the future study.

b) Addition of unimplemented functions

Support for right-to-left (RTL) language is required for internationalized software. We would like to add this functionality in the future. Also, because the supported locales are limited, we consider extending the number of supported locales by seeking volunteers after open-sourcing the software.

## 7. Conclusion

Although software localization (translation) and internationalization (engineering) are a series of processes, they are often done in different locations now. For this reason, translators do not have much chance to check the context while translating UI strings or to understand internationalization. By helping translators to learn about these points, the quality of translation or project management should improve. We developed the localization training software with the aim of providing such education easily. With this software, translators can learn about the translation skills using CAT tools, the scope of internationalization, or different formats in different locales. Although the result of small questionnaire study and comments from the seminar participants show that the software has a certain level of educational effectiveness, further research is needed to determine the effectiveness more precisely.

.....

### About the authors

Ryutaro Nishino: PhD course student at Tokyo Institute of Technology, Graduate School of Decision Science and Technology. Contact: ryutaro@nishinos.com

Kayoko Nohara: Professor at Tokyo Institute of Technology, Graduate School of Decision Science and Technology.

.....

### References

Apple Inc. (n.d.). Internationalization and Localization. Retrieved June 13, 2013, from <https://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/BPIInternational/Articles/InternationalAndLocaliz.html>

Esselink, B. (2000). *A Practical Guide to Localization*. John Benjamins Publishing Company.

Globalization and Localization Association. (n.d.). Localization Definitions. Retrieved June 13, 2013, from <http://www.gala-global.org/view/terminology>

Localisation Research Centre. (2012). 10th Annual LRC Summer School. Retrieved June 13, 2013, from <http://www.localisation.ie/resources/courses/summerschools/2012/index.htm>

O'Hagan, M. (2006). Training for localization (replies to a questionnaire). In A. Pym, A. Perekrestenko, & B. Starink (Eds.), *Translation Technology and Its Teaching* (pp. 39–43). Tarragona, Spain: Servei de Publicacions.

- Ressin, M., Abdelnour-Nocera, J., & Smith, A. (2011). Of Code and Context: Collaboration Between Developers and Translators. *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 50–52). New York, NY,USA: ACM. doi:10.1145/1984642.1984653
- The Unicode Consortium. (n.d.). Language Plural Rules. Retrieved June 13, 2013, from [http://unicode.org/repos/cldr-tmp/trunk/diff/supplemental/language\\_plural\\_rules.html](http://unicode.org/repos/cldr-tmp/trunk/diff/supplemental/language_plural_rules.html)

