Localisation Focus

THE INTERNATIONAL JOURNAL OF LOCALISATION

ISSN 1649-2358

The peer-reviewed and indexed localisation journal

VOL. 13 Issue 1

PUBLISHER INFORMATION

Editor: Reinhard Schäler, *Director*, Localisation Research Centre, University of Limerick, Limerick, Ireland
Production Editor: Karl Kelly, *Manager* Localisation Research Centre, University of Limerick, Limerick, Ireland
Published by: Localisation Research Centre, CSIS Department, University of Limerick, Limerick, Ireland

AIMS AND SCOPE

Localisation Focus – The International Journal of Localisation provides a forum for localisation professionals and researchers to discuss and present their localisation-related work, covering all aspects of this multi-disciplinary field, including software engineering, tools and technology development, cultural aspects, translation studies, project management, workflow and process automation, education and training, and details of new developments in the localisation industry. Proposed contributions are peer-reviewed thereby ensuring a high standard of published material. Localisation Focus is distributed worldwide to libraries and localisation professionals, including engineers, managers, trainers, linguists, researchers and students. Indexed on a number of databases, this journal affords contributors increased recognition for their work. Localisation-related papers, articles, reviews, perspectives, insights and correspondence are all welcome.

To access previous issues online go to http://www.localisation.ie/ and navigate to the Localisation Focus Section

Subscription: To subscribe to Localisation Focus - The International Journal of Localisation www.localisation.ie

Localisation Focus – The International Journal of Localisation (ISSN 1649-2358) is published and distributed annually and has been published since 1996 by the Localisation Research Centre, University of Limerick, Limerick, Ireland. Articles are peer reviewed and indexed by major scientific research services, including: Bowker, Cabell's Directories and St Jerome Publishing Translation Studies Abstracts Online. It is also included in the Library of Congress Collections.

FROM THE EDITOR

All communication is global.

This is a bold statement. But so is it to say: the internet is ubiquitous. If one is true, so is the other.

In this 2014 edition of Localisation Focus – The International Journal of Localisation, researchers report on some of the central challenges that will make global communication across languages possible, and not just for some small number of very profitable "core" languages but for many more of the 6,000 or so underserved languages. Mobile devices are the key that unlocks the door to successful global communication anywhere and anytime – in your language. This is why this edition focuses, in addition to new tools and the shift from "projects" to "services", on tablets and crowdsourcing for app localisation.

For many years, the very active research group at the Faculty of Translation and Interpreting at the University of Geneva in Switzerland, with Maghi King and Susan Armstrong, have lead efforts in the area of MT evaluation and have made significant contributions to its application to localisation scenarios. Lucía Morado Vázquez and Cristina Perón are a new generation of highly active and commited researchers continuing this long-standing tradition in Geneva. Their contribution to this edition, *In-context online localisation tools: the case study of Reverso Localize*, reports on their efforts to develop a framework for the assessment of a new generation of localisation tools, i.e. online tools combined with a machine translation post-editing environment.

The success of tablet computers over the past years has been nothing short of phenomenal. For some time it looked as if desktops would not survive the next decade, now even laptops seem to be on the way out as people increasingly use the smaller, lighter, and more mobile devices. However, while tablets are ideal for content consumption, some users find it difficult to use them for the production of content as they find the performance of even simple tasks like typing more difficult. This is an area that has been investigated by Dr. Gintautas Grigas from the prestigious Institute of Mathematics and Informatics at Vilnius University. Dr Grigas, a long-standing and eminent voice for the rights of the so-called minority languages, especially in the European context, reports on his finding in his contribution to this edition, *Designing Table Computer Keyboards for European Languages*.

While it is sometimes difficult to source specialised localisation training for professionals, one area of training that has been serviced relatively well for some time now is that of localisation project management. At the 12th Annual Localisation Summer School, organised by the LRC in 2014, a student of the MSc in Multilingual Computing and Localisation, Rubén Pérez García, made what I thought was a quite radical proposition: there is no localisation "project" management but rather a localisation "service" management – the localisation industry does not deliver "projects" to their clients, but provides an ongoing "service". He discusses the proposition in his article on *Localisation Service Management Principles*.

Mobile Apps are where markets have developed fastest internationally over the past decade. While even individual developers have managed to create and sell applications on the many different app stores, enough to make a living, the most successful apps have not just made a fortune for their developers, but they have changed our way of communicating and even our way of life – think: WhatsApp or Airbnb. However, according to Susana Muñoz Hernandez and Taygun Bulut Durmaz of the Universidad Politécnica de Madrid, 99% of applications are localised into just 9 languages. In their paper, *Upgrading Mobile Applications Dynamically through Crowdsourcing for Including New Languages*, they describe a strategy and technology framework that will allow the inclusion of many more of the largely underserved 6,000+ languages in the digital world of mobile devices.

I am sure you will find many new and helpful ideas, results, and recommendations in this issue of Localisation Focus. Please keep promoting this journal to your peers. We'd love to hear from you, your colleagues, and your research groups and to report on your work.

Reinhard Schäler

In-context online localisation tools: the case study of Reverso Localize

Cristina Peron, Lucía Morado Vázquez
Département de Traitement Informatique Multilingue
Faculty of Translation and Interpreting
University of Geneva, Switzerland
peroncristina89@gmail.com; lucia.morado@unige.ch

Abstract
Traditional computer-assisted translation tools have been widely used to translate HTML content, and specific localisation tools have also been developed to allow the process of website localisation to be done entirely online with the help of Machine Translation (MT) and by offering an in-context experience. The aim of our work was to evaluate one of the latter tools (Reverso Localize) according to three EAGLES (1996) criteria, namely functionality, usability and efficiency. We carried out an experiment with model-users who localised a website into Italian and post-edited the MT output. The analysis of our results showed that the tool was very responsive and that its two best performing and appreciated functionalities were the interaction system and the in-context post-editing scenario. On the other hand, the usability between other studied functionalities such as the MT output still showed room for improvement. Consequently, we can infer that under specific circumstances this type of tool can represent a real alternative to applying a professional localisation process, but their limitations should also be taken into consideration, and the technical breach that they represent should not be underestimated either.

Keywords: *localisation, website localisation, in-context localisation, online localisation tools, machine translation, in-context post-editing, free localisation software, reverso*

1. Introduction

The rise of the World Wide Web has made distances significantly shorter and has had a considerable impact on market dynamics. Nowadays, a company can sell its products anywhere in the world and its website is very often the first form of introduction to, and interaction with, a potential client (De Bortoli and Maroto 2003). Although English still plays a crucial role in digital communication, several studies have demonstrated that users tend to have a better opinion of a company – that is, they trust and appreciate it more – when they can browse its website in their mother tongue, whether they can understand English or not (Tong in De Bortoli and Maroto 2003). This is the reason that both multinational companies and small and medium enterprises, seeking to expand their business abroad, are increasingly investing into creating multilingual corporate websites (Valdés 2008). Moreover, according to a study by T. Schewe, a company's marketing strategy and its localisation choices are strictly connected (Sandrini 2005). Therefore, website localisation can be considered as "a function of the international marketing strategy" (Sandrini 2005: 4).

Given its significant impact on a company's or an organisation's image, website localisation is a complex process that implies special attention from both a technical and a cultural point of view. As well as traditional computer-assisted translation (CAT) tools, a new line of tools that promises to render localisation an accessible task to web developers has started to emerge. In this study, we evaluate Reverso Localize (from now on RL), an online tool for in-context web localisation, developed by the Reverso-Softissimo team, that promises to enable web owners to have their site localised in several minutes without requiring professional localisers to be involved.

The structure of this paper is as follows: section 2 introduces RL, the tool being studied and offers a general overview of its use; section 3 presents the methodology put into place to evaluate the tool and offers the results of the experiments carried out; our conclusion is presented in section 4.

2. Reverso Localize: motivation and purposes of the research

Reverso Localize is an online localisation platform that uses MT and allows the user to post-edit the raw MT output in context. It was developed by Reverso-

Softissimo as part of the European project Flavius – which started in November 2009 – and was completed and became operational in November 2012. The Reverso-Softissimo team has since been delivering frequent updates and improvements; hence, we must state that all the data contained in this article refers to the platform's development status as of April 2013, when we carried out our research.

Our choice of this platform was especially motivated by the nature of the tool, which allows the post-editor to work in context. In addition, we aimed at testing the real advantages that this type of tool can represent to the web localisation process as well as its weakness and limitations. At the time of our experiment, the free version of the tool allowed the localisation of a website up to 3,500 KB in size (around 30 pages of formatted text) into a maximum of two languages. Twelve languages were available as both source and target and the user could choose between two different translation scenarios: mirror site or file translation. In both cases, a built-in MT engine produced a raw translation of the source text. However, in-context post-editing (PE) was only available when choosing the mirror site option, and so this option was therefore chosen for our study. As indicated by RL developers, the mirror site was mainly aimed at novice users; on the other hand, the file translation was better adapted for more professional localisation processes. The instructions in Figure 1 explain how the mirror site worked and were accessible from the platform:

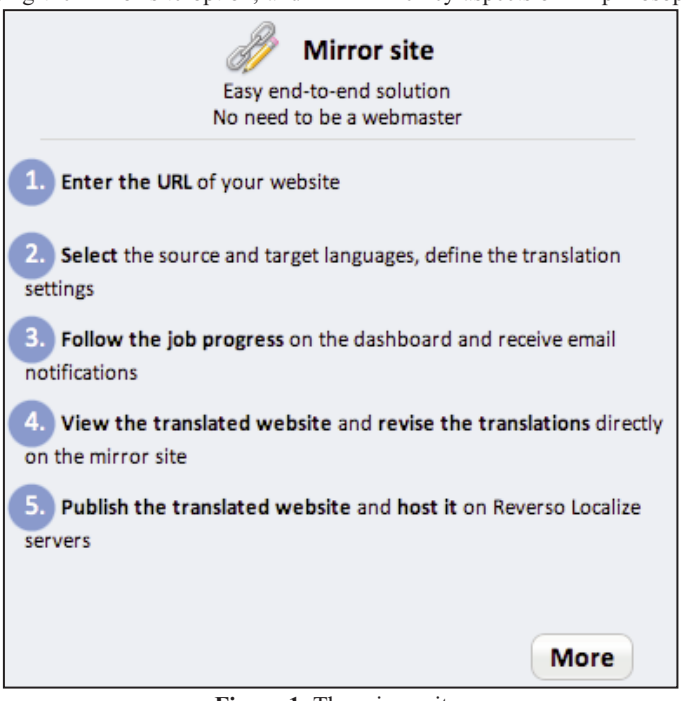

Figure 1. The mirror site.

As explained in Figure 1, a localisation project can be set up quickly by typing the URL of the website to be localised, selecting the language combination and adjusting the settings. The project is managed from the dedicated dashboard. Prior to the translation task, the original content is spellchecked. RL produces a machine-translated version and it then creates a mirror version of the website, that is, a machine-translated copy that can be post-edited in context (see Figure 2). At this point, a post-editor can be invited, or the user himself can decide to proofread the MT output. In the first case, the user and the post-editor work together on the same project within the RL interface, and the user can monitor the PE completion status and the edits. Once the PE is finished, a link provided by RL has to be copy-pasted into the source code of the original website in order to publish and index its localised version. On the platform blog, the user can find instructions on how to publish static and dynamic (Wordpress and OverBlog) sites.

The key aspects of RL philosophy are its ease of use and its speed. Indeed, the software is intended for both professional and non-professional users and the company's challenge is to allow them to localise a website without any prior technical skills and within a very short timeframe. Since this seemed to match

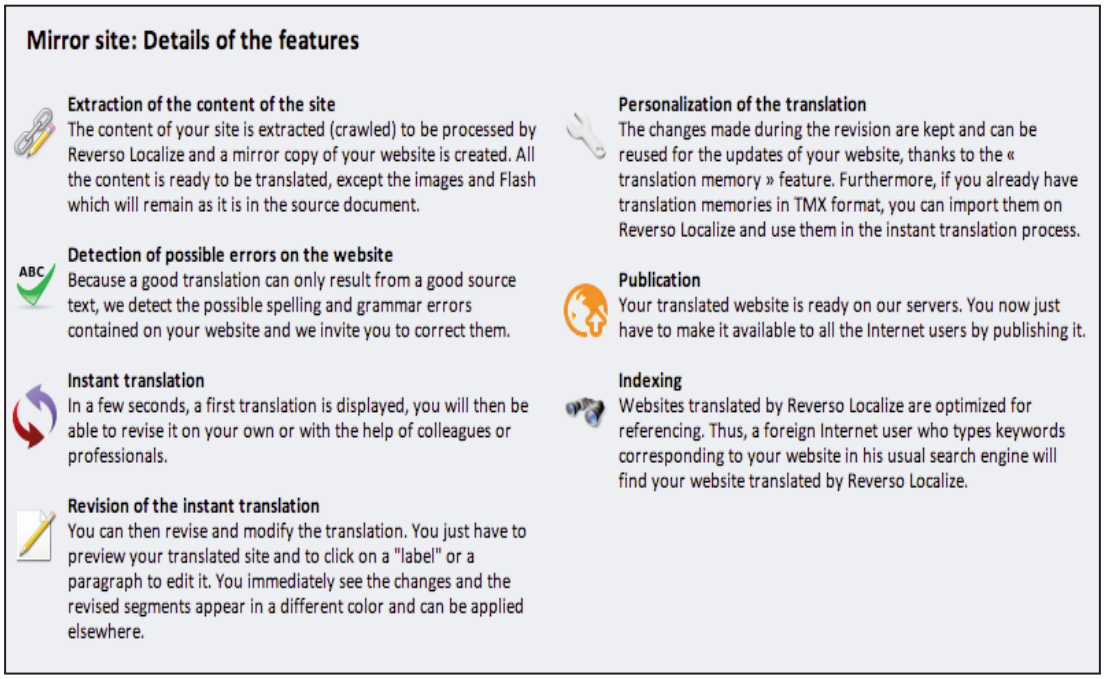

Figure 2. Mirror site: Details of the features.

the current market needs, the main goal of our research was, on the one hand, to estimate the platform's internal performance and, on the other hand, to evaluate user satisfaction. Based on our results, we then assessed the positive aspects of the tool and those that could be improved upon from the point of view of system functionality and system-user interaction. Having said this, a more general goal of ours was to spark a discussion on this new type of tool and lay the groundwork for the creation of an evaluation method.

3. Experimental research

3.1 Methodology

Although the evaluation of translation tools is generally thought of as a very important research field, a dedicated, standard and recognised testing method still has to be developed (Quah 2006). According to Quah, this is due to the plethora and variety of available tools: new systems, featuring diverse functionalities and aiming at different goals are constantly being launched into market. Hence, it is not possible to design a global method that could be applicable to every type of system and, at the same time, that would be detailed enough to be perfectly adjustable to each tool. Since no method met all our requirements – that is, there was no specific method for evaluating a localisation platform with a built-in machine translation engine – we adopted the one developed by the EAGLES (Evaluation of Natural Languages Processing Systems) working group in 1996. This method applies to all language technologies and is based on ISO (International Organisation for Standardisation) 9126 standard. In addition, EAGLES 7-step recipe (EAGLES 1999), describing the seven fundamental steps of an effective evaluation of a natural language processing tool, served as a model for our study.

As recommended by the 7-step recipe, we first defined the motivations and the objectives of our study and then we designed the structure and the parameters of the evaluation, which took a field experiment as its starting point. Eight volunteers participated in it: four users, who worked on the main localisation project and four post-editors, whose task involved adapting the raw MT output. The experiment consisted of localising the Geneva Youth Hostel website (http://www.yh-geneva.ch/) from French into Italian, applying the mirror site scenario. The chosen website was built with the Joomla! content management system (version 1.5). Users worked on the whole French content of the website, whereas post-editors only focused on the machine-translated content of the homepage. All participants were students at the Faculty of Translation and Interpreting (FTI), University of Geneva, Switzerland. Therefore, they were not – or not yet – localisation or post-editing professionals.

The next stage required by the 7-step recipe was to define our own evaluation criteria. We decided to evaluate the platform according to the six criteria proposed by EAGLES (1996) that, in turn, follow the high quality principles established by the ISO 9126 standard: functionality, usability and efficiency. Each criterion was evaluated both objectively and subjectively; which means that our results were derived from three different data analysis methods: first, an objective analysis/extraction, carried out following the principles of systematic test (EAGLES 1996); second, an objective analysis of the material obtained in the scenario test (ibid.); and third, an analysis of the impressions and opinions of our participants (see Figure 3).

EAGLES criterion	Objective evaluation		Subjective evaluation
Functionality	Notification and interaction	-	Questionnaires
	Spellchecker	Precision	
	Localisation system	• Localisation issues • Omissions • Silence	
	Machine translation and post-editing systems	Evaluation of the raw MT output and the post-edited version through SAE J2450 standard	
	Mirror site	-	
Usability	• Mouse click count • Tab count • Keystroke count	Screen recording analysis	
Efficiency	• Internal speed • External speed	Screen recording analysis	

Figure 3. Evaluation methodology based on the 3 EAGLES criteria.

3.2 Workflow, tools and Content

The experiment reproduced a real-life localisation scenario using the platform. It was divided into two parts, respectively devoted to the two main parties involved in a localisation project using RL: users and post-editors. Although we designed a different experimental task for each group, both users and post-editors worked on the same project.

As shown in Figure 4, prior to the main experimental task, all participants had to fill out a general background questionnaire aimed at collecting data on their profile. Data obtained from this questionnaire allowed us to determine that all participants shared similar characteristics in terms of previous experience in web localisation. They were then asked to access the Geneva Youth Hostel website using the Mozilla Firefox web browser and get a first glimpse at its structure and content.

Following this, they had to log into the RL website, read through the home page content and basic information about RL and watch an introductory video in English about the platform. Then, the user needed to take the following steps: log into their account on the platform, type the Gmail address and the related password provided by the researchers; set up a localisation project and adjust the project settings; analyse the source text spelling report,

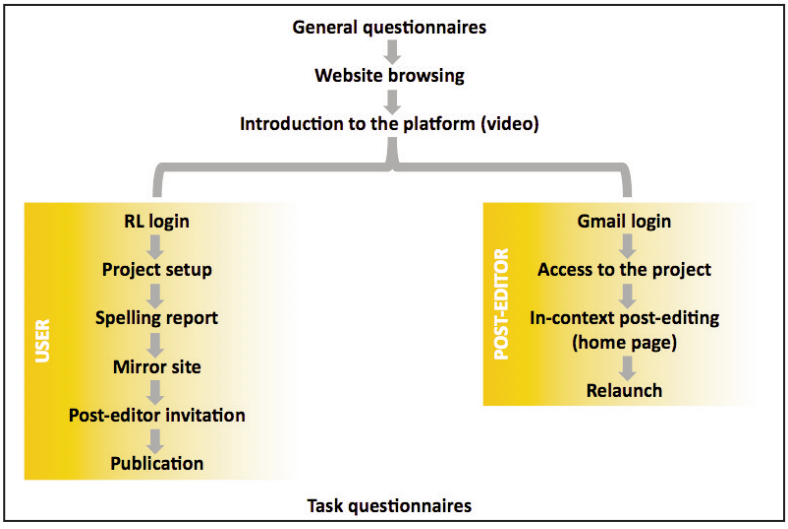

Figure 4. Structure and conduct of the Experiment.

automatically generated by the platform spellchecker; examine the localisation output inside the mirror site; invite a post-editor through the RL invitation system and let them modify the raw Italian MT output. The last task was dedicated to publication and consisted of linking the localised version to the original site. When this experiment was carried out, RL did not provide specific instructions on how to publish a localised version of a Joomla!-based website, as was our case. To overcome this, we drafted a new instruction sheet, taking a document about static sites that was available on the platform as a reference. As a result, users worked offline – as if on a static website – in order to link the localised version to the website code. As the user clicked on the Publish button, RL created a link leading to the localised version of the site. The user then had to download the source code of the original content, open the HTML file with an advanced text editor (Notepad++ in our case) and replace the French code portion with the newly created link, as well as changing "Français" for "Italiano". They also had to replace the French flag icon with the Italian one that was stored in a folder on their desktop. At the end of the process, it was possible to browse the localised version of the website offline.

As for post-editors, after watching the introductory video, they had to log into the Gmail account where they received their invitation. The automatically-

Figure 5. The post-editing system: fixing an incorrect segment.

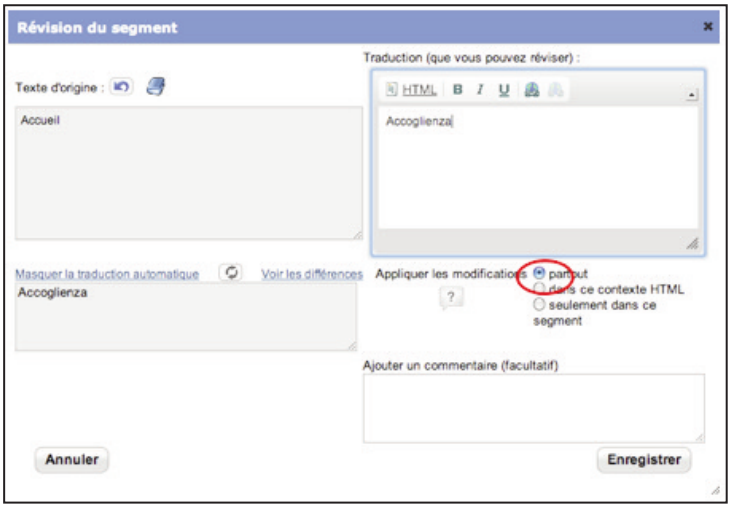

Figure 6. The post-editing system: the post-editing window.

generated email prompted them to click on a link in order to access the localisation project and start proofreading the translated content inside the mirror site. Once there, they were asked to correct the raw MT output in context by clicking on the pencil symbol of any incorrect segment (Figure 5) and editing text and/or links inside a dedicated post-editing window (Figure 6).

When working on RL, the user is constantly notified of any updates and/or editing tasks through email and on-screen notifications displayed on the platform interface. Participants could choose to follow the project status via one of those two notification channels. Once the experiment was completed, they all had to fill in a task questionnaire aimed at collecting their opinion on the platform and its performance.

3.3 Analysis and results

As explained in the methodology description, the three chosen evaluation criteria were verified in both an objective and a subjective manner.

3.3.1 Objective evaluation

The objective analysis of Functionality focused on the performance of the following components and content types: the spellchecker, the localisation system, the raw MT output and the post-edited text. As for the first one, RL provided a spellchecking report of the source text, where errors were classified by type and displayed together with some context and a correction suggestion. We rated the precision of this functionality by calculating the number of real errors contained in the corpus. The spellchecker detected 96 French mistakes on a total 8319 words (data collected on April 14th 2013), only 37 of which were real errors, while the remaining 59 were noise: therefore, the tool scored a 38.5% precision rate.

In order to evaluate the functionality of the localisation system, we concentrated on three types of errors found on the mirror site: localisation problems, omissions (elements that were not present in the localised version) and silences (elements that should have been localised but remained in the original language). We recognised three localisation problems: a character encoding issue (Figure 7, No. 1), two overlapping elements (Figure 7, No. 2) and a truncation issue (Figure 7, No. 3).

As visible in area No. 1, both the character (&) and its entity reference (amp;) were displayed in the localised version. In area No. 2, the overlapping of the magnifying glass and the search bar was due to tabs being longer in the Italian version than in French, therefore taking up almost all of the available space. Finally, in area No. 3, the word truncation issue could had been due to the fact that the Italian segments inside the box were shorter than the original phrases, which probably caused the following text to move back up. RL, however, did not offer a way to fix those three spatial issues to either users or post-editors. As for omissions, we only detected one such error that, nevertheless, affected the whole website: tooltips that are usually displayed on mouseover were not visible in the mirror site and they were neither translated nor shown once the localised site was published. Some recurring localisation problems, instead, were in fact silences. Firstly, it was not possible to edit the dynamic elements such as banner texts or Flash animations as

Figure 7. Localisation Problems.

the tool did not have direct access to the original code of such elements; in a regular localisation process they are usually modified separately with the aid of specific tools, such as advanced image editors. Secondly, a number of website pages that were only accessible via a link on another page (an estimated 30% of total content) were left entirely untranslated. In all likelihood, the reason for this is that RL could only extract data down to a certain depth of the website tree structure. As a result, users browsing the Geneva Youth Hostel localised website were likely to visit one or more subpages containing French text. According to Schiller (2006), such problems can have a high impact on the site credibility, damaging the company-customer relationship and even leading the user to exit the website and never visit it again. This localisation issue was therefore rather serious, especially if we consider that the user could not do anything about it.

As a reference for evaluating MT quality, we chose the SAE (Society for Automotive Engineering) J2450 standard that defines the following seven categories of errors, seen as unacceptable in a translated text: wrong term (WT), syntactic error (SE), omission (OM), word structure or agreement error (SA), misspelling (SP), punctuation error (PE) and miscellaneous error (ME) (SAE J2450 2001). Errors in each category can be classified as Major (_M) or minor (_m).

As shown in Table 1, we detected 48 errors over 28 segments, corresponding to a total of 180 words. Wrong terms were the most represented category (63% of errors), followed by miscellaneous errors (19%), among which we included wrong prepositions and articles, as well as untranslated words. We point out that the system did not make any punctuation or

	WT	SE	OM	SA	SP	PE	ME	TOTAL
_M	19	6	Ø	1	Ø	Ø	2	28
_m	11	1	Ø	1	Ø	Ø	7	20
TOTAL	**30**	**7**	**Ø**	**2**	**Ø**	**Ø**	**9**	**48**
% on total word count	**16,6**	**3,8**	**Ø**	**1,1**	**Ø**	**Ø**	**5**	**26,5**

Table 1. Errors in the raw MT output according to SAE standard.

spelling errors, and we did not identify any omission. Nevertheless, we also note that only 5 segments out of 28 were completely error-free. According to the reference standard (SAE J2450 2001), the error rate is measured with the following formula:

$$\frac{\text{(Error No. 1 x error weight) + (Error No. 2 x error weight) + ...}}{\text{source text word count}}$$

In our particular case, the tool received a score of 0.9, that is, 90% of errors. It is therefore obvious that the raw MT output could have never been published without "full post-editing" (Allen 2003). If we consider the significant editing effort required by the post-editor, this could reopen the debate on the usefulness of machine translation. We will deal with this specific point in the subjective part of the evaluation. We should also state that the MT engine is an external element of RL and therefore it can be changed and improved.

In order to compare MT and PE performance, we also examined the four versions post-edited by our volunteers, taking the SAE standard once again as a reference. The post-edited texts, made up of a total 180 words, contained an average 6 errors, nearly half of which were categorised as minor. It is clear that MT had a great impact on the total error count of the post-edited versions: 12 errors out of 25 were connected to MT. For example, as we observed after examining screen recordings, a post-editor failed to see a specific error twice (but corrected the exact same mistake in other occasions), possibly because he had grown accustomed to seeing the wrong terms displayed on the mirror site: these missed corrections cost him two major wrong term errors, since the raw MT was not acceptable.
For processing MT errors that occurred more than once, post-editors could use the system's relaunch feature in order to apply changes to the whole content of the website. However, this functionality only worked for 100% matches, while other segments – the most part in our case – could only be edited manually. In the same way, we could state that the PE system was quite rudimentary, as it did not provide any functionality for shifting words or for automatically searching and fixing errors.

In order to objectively evaluate the Usability criterion, we made use of BB FlashBack (Express version), a tool developed by Blueberry Software and used for recording activity on a computer screen and extracting useful data for further investigation (e.g., mouse movement, mouse click count, keystroke count and shortcuts used, as well as the time spent on a specific action). To develop the usability criterion, we relied on three aspects related to ergonomics: mouse click count (including scrolling), keystroke and tab count. We calculated the first by working out the total click count, the number of clicks made by each participant during the most important tasks and those required for scrolling and tab switching. The number of clicks performed was similar among all participants and indicated a certain effort level, which was significantly low for some tasks (e.g., interaction between the user and the post-editor while working on the same project) and much more intense for others (project setup, use of notification system, post-editing). This being said, we noticed that the click count was much higher in the experiment with post-editors, while the PE window was the feature that had the biggest impact on the total. Indeed, the PE system did not provide any advanced post-editing features (an easy way to shift words, regular expressions, etc.); this forced the post-editor to make extensive use of the mouse. What is more, a mouse click was also required for validating each translation, as the Enter key could not be used for this. As a result, the quality of the raw MT output and a rudimentary post-editing system were the two elements that had an impact on ergonomic comfort, as low-quality translations required a greater effort from post-editors even from the physical point of view. In our case, we observed that it took them an average of 100 mouse clicks to post-edit a text of 180 words.

During the analysis of recordings, we also calculated the number of tabs opened while working on RL; indeed, having to shift among tabs is likely to have an impact on the total click count, on the amount of time dedicated to a task and on the global ergonomic comfort of the platform. Users opened two RL tabs on average, while post-editors opened five. This was especially due to how RL worked. Indeed, every time that a user ran a certain task, such as accessing the mirror site, relaunching a translation, opening the updated version after relaunching or accessing the latest edits through an email notification, a new tab was automatically opened, even if there was already a tab dedicated to that action. This was detrimental to the platform usability for two reasons. Firstly, working with multiple tabs of the same program could be misleading for the user, this happened to one participant who struggled to find his way back to the tab that he was working on – one of six open tabs – for more than a minute. Secondly, as shown on Table 2, tabs also had an impact on the number of mouse clicks, therefore influencing ergonomic comfort in both ways.

To complete the objective evaluation of Usability, we extracted an XML file containing keystroke and shortcut logging information from BBFlashback. This file gave us an interesting overview of how participants worked, as well as of the RL performing mechanism. We noticed that, for instance, users often resorted to using arrow keys when navigating around the page code, during the publication task. As for post-editors, they mainly used the backspace key for deleting wrong MT solutions and letter keys for editing them, as well as a great number of arrow keys for navigating around the text. These data indicated that tasks having the highest impact on ergonomic comfort were PE and publication.

In order to evaluate the tool's Efficiency, we relied once again on the video recordings collected using BBFlashback. We determined, specifically, the tool's internal speed, namely the average time that it took for RL to complete a given task, and its external speed, that is, how long it took participants to carry out a particular task in the platform. The former was mainly linked to data extraction speed.

At each new project setup, RL carried out four tasks: extracting the data of the original site, spellchecking the source content, translating into the target language and finally displaying the post-editing and the publication windows. Data extraction usually took a short period of time: project setup (including spellchecking, translation and mirror site creation) took an average of 1:13 minutes, while an average of 10 seconds was necessary to create a link that was used to publish the localised version. Looking at these data, one can safely state that the tool's internal speed was undoubtedly one of its major highlights, making it possible for a user to setup a ready-to-post-edit localisation project in slightly more than a minute. If, however, RL failed during project setup – as happened in one of our experiments – the only solution was to start over again, which inevitably reduced ergonomic comfort.

To determine the tool's external speed, we took into account the main tasks carried out by the participants and the time spent on each of them. The collected data about external speed were in line with those on data extraction and we can state that they were not high at all: an average of 1:34 minutes was required for setting up a localisation project, and users needed 34 seconds on average to interact with post-editors who, following this, could work on the project without any additional support. What is more, looking at the total duration of the experiment, we can safely say that all users managed to setup and publish a localised website in less than half an hour (excluding the PE work). The time spent on PE, however, must also be taken into account: it took post-editors 25:51 minutes on average to revise a text that was 180-words long. According to Vasconcellos and León (in O' Brien 2010), the daily output of a professional can range between 4,000 and 10,000 words, when doing "light" post-editing. Since our volunteers were not professional post-editors, we rather chose to refer to other data that set an estimated daily post-editing output of 3,000 to 9,000 words (O' Brien 2010).

Therefore, assuming that they worked 8 hours per day at the pace recorded in our experiment, our post-editors could had processed around 3,320 words per day, a number that is barely acceptable according to O'Brien's estimates.

The publication task, which took 8:43 minutes on average, contributed to increasing the total duration of the experiment. Indeed, we noticed that the participants struggled to complete this assignment. This is no surprise, as having to tinker around with code, deleting and copy-pasting code portions in the editor window, is obviously not a straightforward process. However, we cannot blame RL for this difficulty, since that method was specifically designed by the researchers to tackle a scenario that was not addressed in the platform instructions. Although RL states on its website that no technical knowledge is needed to use the tool, we ask ourselves if it would actually have been possible for a non-professional to figure out how to localise Joomla!-based websites and several other types of sites, for which no instructions were available at the time of the experiment. We can therefore conclude that, even though the platform itself is a responsive tool, working on it is not always a quick job, as it greatly depends on the type of task being performed and the characteristics and nature of the original website.

3.3.2 Subjective evaluation

The subjective evaluation, based on the results of task questionnaires filled out by participants at the end of the experiment, aimed at collecting the opinion of model users about the platform, and their first impressions. Its first part was the same for users and post-editors and focused on the tool's usability and efficiency, as well as on their general satisfaction level. The questions regarding functionality obviously differed, depending on the respondent role: users were asked to rate the spellchecker and the

localisation and publication systems, while post-editors had to give their opinion on the MT results and the post-editing system. Question types were rather varied: Likert scales (where participants expressed their level of agreement or disagreement on an agree-disagree scale for a series of statements) were the most frequent type used, but we also included yes/no, ranking and open questions. Answers to almost all of the questions – excluding the open ones – were measured on a 1-6 scale, 6 being the highest or best score.

As far as Usability is concerned, RL scored a total average score of 4.2. Participants were particularly satisfied with the very low learning effort required, the effectiveness of instructions and the interface, which they described as simple and easy to use. However, they showed some perplexity about whether they could have completed all the tasks without the instruction sheet provided, and they gave the corresponding question a lower score (3.75).

To evaluate Efficiency, users and post-editors were asked if the time spent on localisation and post-editing was acceptable and if they thought that RL could allow a localiser or post-editor to save time. The final score for this criterion was 4.4. According to participants, the time spent on their tasks was acceptable, or even short for one of them. All users agreed that RL allowed localisers to save time, every one of them giving positive scores and an average score of 5; post-editors, however, gave a much lower average score of 3. This result can be explained by linking it to the Functionality criterion and, particularly, to the respondents' answers on the raw MT output. There, they stated that the MT slowed their work progression down and that, in general, they would have preferred to translate from scratch. In other words, they thought that the time spent working on the platform was by and large acceptable, but the tool did not allow for a significant time-saving.

Among functionalities, the best rated one was the mirror site (average: 5.5), which was described as useful, easy to use and innovative, and received only positive scores. The "Notifications and interaction" functionality also scored well, especially for allowing users and post-editors to interact with each other, collaborate on the same project and see the edits in real time. The notification system, on the contrary, was criticised on some points. As already said, every time that a user completed an action, he/she was notified both via email and on-screen within the RL interface. The user could not configure nor modify those default parameters. However, half of participants declared that, if they could have chosen one or the other, they would have only kept the on-screen notification system activated. From these answers, we can infer that participants may not have liked receiving an email notification for something that they could have checked within the RL interface itself. All the more so as checking one's inbox meant having to open multiple browser tabs and losing concentration.

The localisation system received an average score of 4. Participants found it efficient (causing few issues in the localised version) and well-structured. However, users were less enthusiastic when talking about the publication task; although they did not consider it too challenging, they stated that it would have taken them longer – or several attempts – to complete it without the instruction sheet designed by the researchers. Another element that scored quite poorly was the page footer that contained the platform logo, flags for changing the language and a string saying: "This website has been automatically translated by the Reverso Localize platform". Post-editors pointed out that their role was omitted by this sentence and they found it unfair that it gave all the credit to Reverso MT system when, in fact, its output was edited by a human.

The spellchecker received a 3.8, which was slightly below the pass score. Users noted that the real errors-noises ratio was reasonable, the relevance of correction suggestions was acceptable and that, by and large, this latter functionality could be quite useful.

Finally, MT scored an average of 2.8. Users who evaluated the performance of the system gave it an average of 1.75 and claimed unanimously that MT slowed down their work rate, the majority of them saying that they would have rather translated the source text from scratch. We should state that comparing MT+PE against translation from scratch did not fall under the scope our research and therefore we did not do any empirical test in that sense; but it would be an interesting path to follow in the future, especially after having heard our participants opinions on that topic.

4. Conclusions

On the basis of our study and the collected data, we will now try to draw some conclusions regarding the localisation platform. On the whole, we can state that the intended goal was accomplished: non-

professional users had localised/post-edited a website in less than half an hour using RL.

Let's now have a look at the most effective features of the system and those that could be improved. Among the evaluated functionalities, the post-editing environment (that is, the mirror site) was the best performing as both users and post-editors found it useful, straightforward and, above all, innovative. Indeed, the in-context post-editing scenario brought by RL introduced a groundbreaking concept in the panorama of existing CAT/localisation tools on the market. As for the interaction system, it was equally appreciated by participants thanks to its high cost-effectiveness.

The localisation system still needed to be improved on some points: although users appreciated its approach, they also identified some irreversible errors that hinder the tool's performance. A user-system interaction option should be introduced in order to let users fix interface issues (for instance, the platform could incorporate a field and window resizing feature or a method for translating possible silences). What is more, the publication task was not yet exhaustively documented on the platform; instructions were not available for every website type, therefore affecting non-professionals' ability to work independently. The spellchecker was another tool with mediocre performance. Despite being well-structured and easy to use, it only achieved a 38.5% precision rate. On the other hand, technically speaking, in RL the spellchecker is an external element, which means that RL can easily incorporate another spellchecker with a better performance if required.

MT and PE systems were, however, the less performing features of the platform. The reason being that the target text required a thorough and continuous intervention from the post-editors, who lacked access to functionality that could increase their work rate and complained that it took too many mouse clicks to edit each segment. Participants gave their lowest – and well below the pass – scores to the MT system. The internal functions of the software should be improved, for instance, allowing users to customise the dictionaries. In a similar way, it would be important to reinforce the PE system with a word-shifting feature, as well as automatic search patterns and error correction. As matters stand now, both the MT and PE systems seemed to have a negative impact not only on the tool's global performance, but also on its usability and efficiency.

As for Usability, we can draw two conclusions. Due to its structure and the way it was designed, the platform implied a considerable effort in terms of necessary mouse clicks and tab shifting. On the other hand, RL required a low learning effort as instructions were clear and the interface was straightforward.

The Efficiency of the tool can be considered one of its main advantages. The platform turned out to be noticeably fast; its external speed was quite good for some tasks, such as project setup and interaction, but it was not as optimal on others, such as PE and publication.

Even though the evaluated tool was an innovation in its field, and it was therefore difficult to compare it with other existing software, or to collect data about future trends, we can conclude that RL came as a real novelty, featuring key points like the mirror site, the interaction and localisation systems, and the cost-effectiveness of the program.

The downsides to the program were the MT output, the PE system and some aspects of its usability. If those last aspects were to be improved, RL would become a very competitive program, taking into account that it is free and easily accessible. However, as with everything web-related it is likely to change at a fast pace. Indeed, as of the date of the publication of this article, several updates and modifications have been made to the software, compared to when the study was performed. An interesting direction for future investigation would be to evaluate the tool once again, in order to measure the impact of the updates and compare the new results with those of this study. It would also be useful, then, to increase the number of participants and repeat the experiment with different parameters of the tool, to see if the trends differ.

Hence, RL is quite unique in its kind, and it integrates various technologies. Consequently, we could not base our study on a specific model or standard designed for such a tool, nor could we take previous works as a reference, or compare it to similar platforms. This work constitutes, therefore, an introductory investigation, the results of which could, in the future, be compared to those of other types of tools. Finally, it could also become a blueprint for evaluation, because it has allowed for the narrowing down of the functionalities that are vital for these tools.

Acknowledgements

This work was carried out in 2013 under the supervision of Prof. Pierrette Bouillon and its extended version was defended as the Master thesis of Cristina Peron at the University of Geneva.

We would like to acknowledge all the other individuals that collaborated with this research: the Reverso Localize team for their assistance and support; the experiments' participants who donated their time; and Miss Mary Sue Giachetti, for her help in the editing process.

References

De Bortoli, M., Maroto, J. (2003) *Web Site Localization* [online], available: http://pure.au.dk/portal/files/11487/Appendix_3.pdf [accessed 30 May 2013]

EAGLES Evaluation Working Group (1996) *Evaluation of Natural Language Processing Systems - Final Report* [online], available: http://www.issco.unige.ch/en/research/projects/ewg96/index.html [accessed 30 Jan 2013]

EAGLES Evaluation Working Group (1999) *The EAGLES 7-step recipe* [online], available: http://www.issco.unige.ch/en/research/projects/eagles/ewg99/7steps.html [accessed 30 Jan 2013]

O' Brien, S. (2010) *Introduction to Post−Editing: Who, What, How and Where to Next?* [online], available: http://amta2010.amtaweb.org/AMTA/papers/6-01-ObrienPostEdit.pdf [accessed 12 Jun 2013]

Quah, C. K. (2006) *Translation and technology*, Basingstoke: Palgrave Macmillan

Reverso-Softissimo (2013) 'Comment ça marche ?', *Reverso Localize* [online], available: http://localize.reverso.net/HowItWorks.aspx?lang=fr [accessed 15 Jun 2013]

Sandrini, P. (2005) 'Website localization and Translation' in Gerzymisch-Arbogast, H., Nauert, S., eds., *Challenges of Multidimensional Translation - Proceedings of the Marie Curie Euroconferences MuTra*, Saarbrücken, 2-6 May 2005 [online], available: http://www.euroconferences.info/proceedings/2005_Proceedings/2005_proceedings.html [accessed 28 May 2013]

Schiller, A. (2006) *Website translation: issues of cohesion* [online], available: http://www.unizar.es/aelfe2006/ALEFE06/4.translation/74.pdf [accessed 16 May 2013]

SAE J2450: *Translation quality metric* (2001) Warrendale: SAE International

Valdés, C. (2008) 'The localization of promotional discourse on the internet' in Chiaro, D., Heiss, C., Bucaria, C., eds., *Between Text and Image: Updating research in screen translation*, Benjamins Translation Library, p. 227-240

Designing Tablet Computer Keyboards for European Languages

Dr. Gintautas Grigas
Vilnius University Institute of Mathematics and Informatics,
Vilnius, Lithuania
www.mii.vu.lt
gintautas.grigas@mii.vu.lt

Abstract

The keyboard of a tablet computer is on-screen and programmable. There are no restrictions on the number of keys. The keyboard programmer has the freedom to design a keyboard in accordance with the alphabet of a particular language. This paper analyses alphabets of different languages using the Latin script and presents the classification of letters according to their usage, frequency, attitude to main (native) language and foreign languages. It discusses methods of designing tablet keyboards for different languages using the experience of the English language. It presents a way to minimize the area that the keyboard occupies on the screen. It looks at direct typing as the only suitable typing method for all the letters of the main language, including all language-specific letters, and the press-and-hold typing method, which ensures a uniform way to get many extra symbols and allows for the development of international keyboards that include letters capable of covering the alphabets of a number of foreign languages.

Keywords: *on-screen keyboard, sensory keyboard, international keyboard, keyboard design, language-specific letter, letter frequency, ĄŽERTY, QWERTY, press-and-hold, tablet computer.*

1. Introduction

The most important factors in the design of a keyboard are a set of symbols and their layout. The number of letters in the alphabet varies from language to language. The number of keys in the physical keyboard is fixed. The keyboard of the tablet computer is on-screen and programmable. We can "draw" as many keys as necessary.

In the case of physical keyboards, we are forced to fit the alphabet to the keyboard. In the case of on-screen keyboards, we may adjust the keyboard to the alphabet in order to make it more natural for the language.

Another useful property of on-screen keyboards is the ability to access extra letters (characters) via the press-and-hold method. This method involves programming the keyboard so that, after pressing and holding a letter key, a bar appears next to that key with new letters to choose from. These are usually derived from the pressed letter by adding diacritics. It is like adding new temporary keys.

The screen of a tablet computer is usually small. The size of keyboard is not strictly limited, but it is necessary to make design the keyboard in such a way that there would be sufficient space on the screen for text typing and some human-computer interface elements.

With a physical keyboard, text is usually typed while the keyboard is placed on a desk, a lap or on some other horizontal surface. Both hands, and all fingers, can be used for typing and this allows for increased productivity via typing speed. This is an objective of ergonomic keyboard layout design.

A tablet computer or mobile device with an on-screen keyboard is usually held in the hands. Therefore users often have to type with one or two finger. As a result, productivity decreases.

Alternate methods of typing using on-screen keyboards raises doubts about the traditional QWERTY layout. Different layouts of the English alphabet were proposed by MacKenzie, Zhang, and Soukoreff (MacKenzie, Zhang 1999; MacKenzie, Soukoreff 2002).

Less attention is paid to keyboards of other languages. It was only in 2012 that Bi, Smith and Zhai presented the English, Spanish, French, German, and Chinese (pinyin) original letter layouts, as well as a joint five-language keyboard with 49 letters. It is difficult to fit so many keys into a limited area on the screen and so it is necessary to reduce their size. The authors have presented and compared two variants:

> *A separate key is dedicated for each letter and each letter is obtained by a single keystroke;*

> *Separate keys are assigned only to letters without diacritics and other letters are obtained by additional keystrokes or by the press-and-hold method.*

Here we discuss the design principles of tablet computer keyboards that are common for all languages using the Latin script. We start from an analysis of alphabets. According to typing method we divide letters into two sets (groups): primary set (the letter appears immediately after pressing its key) and secondary set (the letter is obtained by the press-and-hold method). According to MacKenzie and Tanaka-Ishii (2007) if each symbol is assigned to a dedicated key, it has no ambiguity. Thus, the primary set of letters does not create ambiguity.

Naturally, the main language letters are included in the primary set, and the letters of other languages are included in the secondary one. These secondary letters may be necessary in dealing with neighbouring countries, typing foreign personal names, names of foreign institutions, and so on.

For illustration we will use the Android keyboard as our default keyboard since this operating system has the largest share, 62%[1], of the tablet computer market.

2. The Current Situation

Currently, in the Android operating system, all letters have primary status for approximately 50% of languages, mainly those having a small number of language specific (those which are absent in English alphabet) letters (e.g. Danish, Estonian, Finnish, German, Swedish). In the keyboards of other languages, only the English alphabet is provided. A similar situation exists in other operating systems as well. So for many languages it is suggested to type the language-specific letters as secondary letters. In order to type a secondary letter, the following actions should be taken:

touch the key of a primary letter
hold it until a bar appears with the secondary letters
find the necessary letter on the bar and touch to select it

This constitutes two key taps and a pause between them each tap. The pause takes approximately 250–500 ms (Bi et al. 2012). This is equal to the time of one key touch when the typing rate is 2–4 characters per second or 24–48 words per minute (such numbers are provided by many authors). Thus, the typing time of one secondary letter is approximately equal to the typing time of three primary letters.

The cultural aspects are important as well. “Product designers should also consider how native speakers conceptualize diacritic characters” (Bi et al. 2012).

This situation deserves correction. First we will discuss which letters should be in both letter sets: primary and secondary.

3. The set of primary letters

The most important parameter in determining the on-screen keyboard design is the number of letters in the alphabet. The number of letters varies in different languages. Furthermore, in some languages there are some rarely used letters and the question arises as to whether they should be included in the primary or secondary set.

We rely on the classification of letters into two types A and B according to the ETSI standard 202 230 (2007), meant for sets of symbols, sorting and layout of 12 keys (T12) in the mobile phone keyboards:

> Type A. Letters essential for the language.
> Type B. Letters used in writing the language, but not essential for it, e.g. used for foreign words and/or spelling of some names in the country where the language is used.

The letter distribution into types A and B for all European Union official languages and several other languages (Icelandic, Norwegian, Serbian, Turkish), using the Latin script, is illustrated in Table 1. Comments on the letters of type A are not included into the language-specific set

1 **English *À Æ Ç É È Ê Ë Ï Ñ Ô Ö Œ***. In the English language there are loanwords with these letters, for example, café, naïve. They are not included into the language-specific set whereas they are used only in foreign words and are absent in English keyboards.
2 **Estonian *Š Ž*.** They are officially listed in the Estonian alphabet but only to write personal names of other nationalities.
3 **Irish *À Æ Ç È Ë Ê Ï Ñ Ö Ô Œ*.** These letters are used in order to write words of other languages.
4 **Latvian *Ō Ŗ*.** After the Latvian language reform in 1946 the letter Ŗ was abandoned, and later the letter Ō.

Language	Lang. code	No. basic Latin letters		Language-specific letters		Number of letters		Comments
		Type B letters	*a*	Type A letters	*b*	*c*=26+*b*	*d*=*c*-*a*	
Czech	cs		0	Á Č Ď É Ě Í Ň Ó Ř Š Ť Ú Ů Ý Ž	15	41	41	
Danish	da		0	Æ Ø Å	3	29	29	
German	de		0	Ä Ö ß Ü	4	30	30	
English	en		0	(À Æ Ç É È Ê Ë Ï Ñ Ô Ö Œ)	0	26	26	1
Spanish	es	K Y	2	Á É Í Ñ Ó Ú Ü	7	33	31	
Estonian	et	C Q W X Y	5	Ä Ö Õ Ü (Š Ž)	4	30	25	2
Finnish	fi		0	Ä Ö	2	28	28	
French	fr		0	À Â Ç É È Ê Ë Î Ï Ô Œ Ù Û	13	39	39	
Irish	ga		0	Á É Í Ó Ú (À Æ Ç È Ë Ê Ï Ñ Ö Ô Œ)	5	31	31	3
Croatian	hr	QWXY	4	Č Ć Đ Š Ž	5	31	27	
Hungarian	hu	Q	1	Á É Í Ó Ö Ő Ú Ü Ű	9	35	34	
Icelandic	is	C Q W Z	4	Á Đ É Í Ó Ú Ý Þ Æ Ö	10	36	31	
Italian	it	W Y	2	À É È Ì Ó Ò Ù	7	33	31	
Latvian	lv	QWXY	4	Ā Č Ē Ģ Ī Ķ Ļ Ņ Š Ū Ž (Ō Ŗ)	11	37	33	4
Lithuanian	lt	QWX	3	Ą Č Ę Ė Į Š Ų Ū Ž	9	35	32	
Maltese	mt	C Y	2	Ċ Ġ Ħ Ż	4	30	28	
Dutch	nl		0	(Ä Ë Ï Ö Ü)	0	26	26	5
Norwegian	no		0	Æ Ø Å (Æ É)	3	29	29	6
Polish	pl	Q V X	3	Ą Ć Ę Ł Ń Ó Ś Ź Ż	9	35	32	
Portuguese	pt	K W Y	3	Á À Â Ã Ç É Ê Í Ó Ô Õ Ú (Ü)	12	38	35	7
Romanian	ro	Q W Y	3	Â Ă Î Ş Ţ	5	31	28	
Swedish	se		0	Å Ä Ö	3	29	29	
Slovak	sk	Q W X	3	Á Ä Č Ď É Í Ĺ Ľ Ň Ó Ŕ Š Ť Ú Ý Ž	16	42	39	
Slovenian	sl	Q W X Y	4	Č Š Ž	3	29	25	
Serbian	sr	Q W X Y	4	Č Ć Đ Š Ž	5	31	27	
Turkish	tr	Q W X	3	Ç Ğ İ Ö Ş Ü	6	32	29	
Average			1.9		6.5	32.5	30.6	

Table 1. Letter distribution based on the standard ETSI 202 230.

5 **Dutch *Ä Ë Ï Ö Ü*.** These letters are used in order to write words of other languages.
6 **Norwegian *Æ É*.** These letters are used in order to write words of other languages.
7 **Portuguese *Ü*.** This was used in Brazilian Portuguese until 2008. Since 2009 it is permitted only in loanwords and personal names.

Basic Latin letters play an exceptional role. Almost all of them are included in the Latin script alphabets and fall into type A. Instead of a long list of them in the column *type B letters* we use a short list of letters that fall into type B. So this column provides two types of information: which and how many letters (*b*) fall into type B, which letters and how many letters (*26-b*) fall into type A. As a result letters-candidates for the primary set were divided into three groups:

1 Basic Latin letters present in their alphabet.
2 Language-specific letters present in their alphabet.
3 Basic Latin letters absent in their alphabet.

That the first group is included in the primary set is undisputed. A part of the second group of letters is written in parentheses. These letters are attributed to type A in the ETSI standard, but rarely used in the main language. Therefore, they should not be included in the primary set. We will discuss the inclusion of letters from groups 2 and 3 into the primary set separately.

4. Language-Specific Letters

Language-specific letters are an inseparable part of the alphabet and so they are in the primary set. In personal and mobile computers they are primaries. Exceptions – several rarely used letters may be moved to the secondary set and typed with a few keystrokes. This is because of the limited number of keys on the physical keyboard. The on-screen keyboard does not have this restriction. However, as already mentioned, in the keyboards of many languages on tablet computers, all of the language-specific letters are considered as secondary letters. Why?

One of the reasons is probably the historical context, inherited from the mobile phones. The number of physical keys on early phones was fixed at 12. Several letters may be accessed via a single key with a different number of keystrokes. The ETSI standard has set the following ordering of letter groups for a key:

- basic Latin letters;
- digit;
- other Latin letters.

For example, in the German keyboard multiple presses of the key 2, will cause characters to appear in the following order: *abc2äàá.*

A digit can be considered as a marker that divides letters into two parts: before the digit (primary set) and after the digit (secondary set). That is why the native German letter *ä* falls into the same group as the foreign letters *àá*.

For a letter that is after the digit an additional press is necessary to skip the digit. This fact causes typing slowdown, supports unmotivated differentiation of the letters of the main language, and does not correspond to the distribution of the letters to types A and B of the standard itself.

Similarly the phenomenon of ignoring language-specific letters has migrated to mobile phones possessing physical alphabetic keyboards with a fixed number of 26 keys for 26 English letters. The next migration step leads to the on-screen keyboard, where there are no restrictions to the number of keys.

There exists an opinion that the frequencies of letters with diacritics are small, not higher than the frequency of any English letter, and the slowdown in typing speed that they cause will have no noticeable effect on the overall text typing speed (Bi et al. 2012). In fact, many of them are at the end of the list ordered by frequency. Here are some examples of letters ordered by frequencies (Letter & Word Frequencies 2014):

Spanish: e a o s r n i d l c t u m p b g y **í** v q **ó** h f z j **é á ñ** x **ú ü** w k;
Polish: i a e o z n s c r w y **ł** d k m t p u j l g **ę** b **ą** h **ż ś ó ć ń** f **ź** v q x;
Swedish: e a n t r s l i d o m g k v **ä** h f u p **å ö** b c j y x w z é q;
German: e n i s r a t d h u l c g m o b w f k z v **ü** p **ä ß** j **ö** y q x;
French: e s a i t n r u l o d c m p **é** v q f b g h j **à** x **è** y **ê** z **ç ô ù â û î œ** w k **ï.**

We see that the highest frequency of language-specific letters is not insignificant: French é 1.5%,

German *ü* 1%, Polish *ł* 2.11%, Finnish *ä* 3.58%, Danish *å* 1.19%, and Swedish *ä* 1.8%. They should not be placed in the secondary set.

It remains to examine some of the least used letter frequencies. Let us compare them with the English letter frequencies (Fig. 1).
The most rarely used English letter is Z, and its

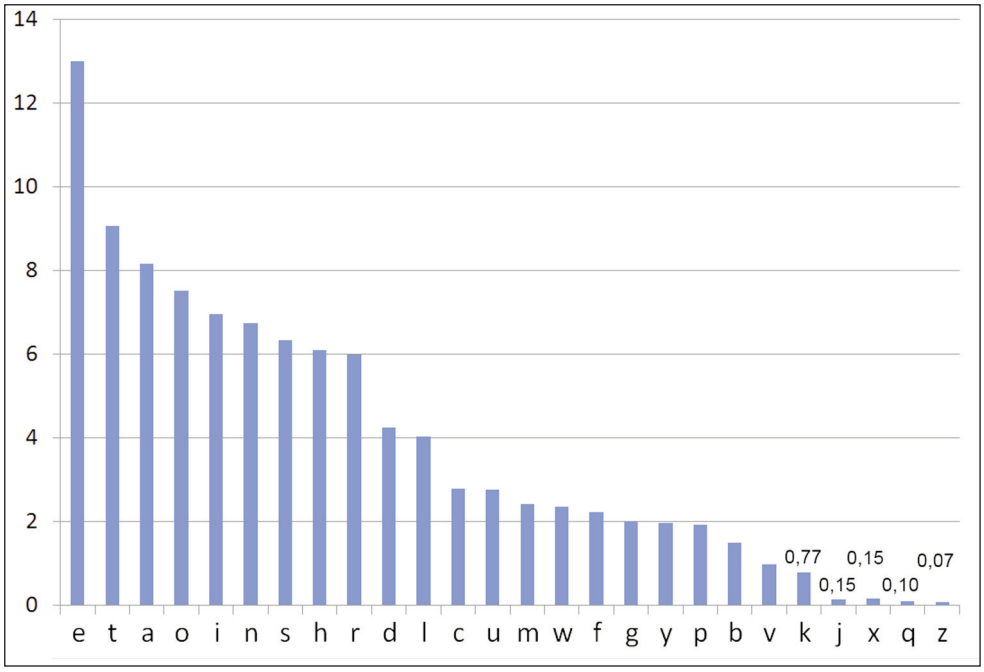

Figure 1. Frequency (in %) of English letters (Lewand 2000, Algoritmy.net 2014).

frequency is 0.07%. Our survey shows which language-specific letters of other languages are below this value:

French:	â 0.051%, œ 0.018%, ë 0.001%, î 0.045%, ï 0.005%, ô 0.023%, ù 0.058%;
Italian:	0.03%, ò 0.0002%;
Spanish:	ü 0.012%;
Finnish:	å 0.003%;
German:	none;
Portuguese:	none;
Turkish:	none;
Swedish:	none;
Polish:	none;
Danish:	none;
Icelandic:	none;
Slovenian:	none.

We see that there are few language-specific letters whose frequency is lower than that of any English letter. More of them, 7, are in the French language only. However, 6 French letters whose frequencies exceed that of some English letters remain.

5. Basic Latin letters that are absent in the main language alphabet

In many languages (see Table 1) there are some basic Latin letters that not used. In fact, they mainly come into play when there is a need to write foreigners' names and other foreign words. In addition, they are used in special computer texts: command names, computer component names, settings, and so on. Thus, their priority should be higher than other letters used in foreign names and foreign words, but lower than the language-specific letters.

Input texts for interaction (i.e. typing commands, settings, etc.) are of little use with tablet computers and so the usage of the aforementioned letters is lower than in desktop computers. Thus, on tablets with small screens it may be rational to bring these letters to the secondary set.

6. Keyboard layout design

A separate on-screen keyboard can be "drawn" for

each language. However, there are some elements common to many languages.

Keyboards for languages with a small number of language-specific letters already exist. Keyboards for languages with more letters have not been developed yet. This might be due to designer fears that keys will become too small if the number is increased. However, in many existing keyboards there is a small reserve that appears in the bottom row (with the spacebar) when the number of keys in letter rows is increased. This reserve usually serves to widen the space key or other keys in its row (Grigas 2014).

Let us begin with the usual English keyboard with 26 letters in 3 rows (Fig. 2). We do not investigate the keys in the bottom row and leave them unlabelled. The physical length of rows remains the same when new keys are added, because the screen size remains the same. However, a relative length, measured by the number of keys, increases, because the keys become smaller. After adding *k* keys to each letter row we gain room for *k* keys in the bottom row as well. Here the punctuation marks can be moved from the letter rows, and the remaining *k* blank keys in the letter row might be used for additional letters. Placing punctuation marks on the bottom row is natural for tablet computers, as there are already keys for such a category.

If k = 1 then we get 4 extra keys in letter rows, together 26 + 4 = 30 keys.

If k = 2 then we get 8 extra keys in the letter rows, 34 keys altogether (Fig. 3). This number of keys is closest to the average number of the letters of the European languages (32.5, see Table 1). So it is reasonable to use this variant as a start point for investigation or design of European keyboards.

If k = 3 then we get 11 extra keys in the letter rows and one extra key in the bottom row, which can be used for one more letter (if there are 38 letters in the alphabet) or additional punctuation marks.

This method of getting additional keys in the bottom row is universal and suitable for keyboards in different operating systems.

Figure 2. Initial keyboard template.

Let us take the following two keyboard examples. The Lithuanian ĄŽERTY keyboard has 9 language-specific letters (Fig. 4). Its layout corresponds to the standard LST 1582:2012, except for the fact that the letter *x* is lowered one row down from the row of digits.

A space for an extra key in the row *asd*...was obtained by deleting the indent of this row and

Figure 3. The keyboard widened by two keys in each row (k = 2).

Figure 4. Lithuanian keyboard (k=2).

shortening the Enter key (compare with Fig. 2).

There are no letters Q, W and X in the Lithuanian alphabet. The ĄŽERTY keyboard has a unique feature in that these letters are at the rightmost end of rows and it is convenient to reduce the keyboard, for example, when the screen of a device is very small. After removal of the right column of letters and shortening the key underneath it, the positions of the rest of the letters remain unchanged.

The Latvian alphabet contains 33 letters (see Table 1). Together with the basic Latin letters Q, W, X, Y that are absent in the Latvian language, it grows to 37 letters. It exceeds the number of available keys in physical PC keyboard, and these four letters were not included in the Latvian standard (LVS 23-93). There is room for all letters in the on-screen keyboard (Fig. 5).

Figure 5. Latvian keyboard (k = 3).

11 extra keys can make a complete row. In such cases it might be reasonable to add the fourth letter row instead of widening three existing rows.

7. International keyboards

The press-and-hold method is a convenient way to access letters of other languages and to make an international keyboard. A bar with a number of letters appears when keeping a key pressed. The number of letters on the bar is not strictly limited. This is the advantage of the press-and-hold method. It is suitable for the international keyboard covering many languages, e.g. all the official European Union languages using the Latin script.

Table 2 shows a list of letters of the EU languages and some other languages, using the Latin script. We assume that in the set of primary letters all letters of the main language and, optionally, all basic Latin letters are included. We also assume that the set of secondary letters will include everything that is left from the primary set, i.e. a union of the alphabets of all languages minus the primary set of a particular language.

The overall convenience of the typing experience and the speed of accessing language specific letters via the press-and-hold method depends on its position on the bar. The closer to its primary letter key, the better. As such locating foreign letters on the bar according their frequency in the main language is the most useful and preferred solution.

8. Conclusions

As this paper has shown, programmable on-screen keyboards are more adaptable for alphabets with

Letter	Unicode		Languages
	A	a	
Á á	00C1	0.00E+00	cs es ga hu is pt sk
À à	00C0	0.00E+00	fr it pt
Ă ă	102	103	ro
Å å	00C5	0.00E+00	da fi no se
Ä ä	00C4	0.00E+00	de et fi se sk
Ā ā	100	101	lv
Â â	00C2	0.00E+00	fr pt ro
Ã ã	00C3	0.00E+00	pt
Ą ą	104	105	lt pl
Æ æ	00C6	0.00E+00	da is no
Č č	010C	010D	cs hr lt lv sk sl sr
Ć ć	106	107	pl, sr
Ċ ċ	010A	010B	hr mt
Ç ç	00C7	0.00E+00	fr pt tr
Ď ď	010E	010F	cs sk
đ Đ	110	111	hr sr
Ð ð	00D0	00F0	is
É é	00C9	0.00E+00	cs es fr ga hu is it pt sk
È è	00C8	0.00E+00	fr it
Ë ë	00CB	00EB	fr
Ě ě	011A	011B	cs
Ė ė	116	117	lt
Ē ē	112	113	lv
Ê ê	00CA	00EA	fr pt
Ę ę	118	119	lt pl
Ģ ģ	122	123	lv
Ġ ġ	120	121	mt
Ğ ğ	011E	011F	tr

Table 2. Language-specific letters of the official European Union languages using the Latin script, and Icelandic, Norwegian, and Turkish languages - Part 1.

Letter	Unicode		Languages
	A	a	
Ħ ħ	126	127	mt
Í í	00CD	00ED	cs es ga hu is pt sk
Ì ì	00CC	00EC	it
Ī ī	012A	012B	lv
Î î	00CE	00EE	fr ro
Ï ï	00CF	00EF	fr
İ ı	130	131	tr
Į į	012E	012F	lt
Ķ ķ	136	137	lv
Ł ł	013B	142	pl
Ļ ļ	013B	013C	lv
Ĺ Í	139	013A	sk
Ľ ľ	013D	013E	sk
Ñ ñ	00D1	00F1	es
Ň ň	147	148	cs sk
Ń ń	143	144	pl
Ņ ņ	145	146	lv
Ö ö	00D6	00F6	de et fi hu is pt se tr
Ó ó	00D3	00F3	cs es ga hu is it pl pt sk
Ò ò	00D2	00F2	it
Ø ø	00D8	00F8	da no
Õ õ	00D5	00F5	et pt
Ô ô	00D4	00F4	fr pt
Ő ő	150	151	hu
Œ œ	152	153	fr
Ř ř	158	159	cs
Ŕ ŕ	154	155	sk
Š š	160	161	cs hr lt lv sk sl sr

Table 2. Language-specific letters of the official European Union languages using the Latin script, and Icelandic, Norwegian, and Turkish languages - Part 2.

Letter	Unicode A	a	Languages
Ś ś	015A	015B	pl
Ș ș	218	219	ro
Ş ş	015E	015F	tr
ß		00DF	de
Ť ť	164	165	cs sk
Ț ț	021A	021B	ro
Þ þ	00DE	00FE	is
Ú ú	00DA	00FA	cs es ga hu is pt sk
Ù ù	00D9	00F9	fr it
Ü ü	00DC	00FC	de es et fr hu tr
Ů ů	016E	016F	cs
Û û	00DB	00FB	fr
Ū ū	016A	016B	lt lv
Ű ű	170	171	hu
Ų ų	172	173	lt
Ý ý	00DD	00FD	cs is sk
Ÿ ÿ	178	00FF	fr
Ž ž	017D	017E	cs hr lt lv sk sl, sr
Ź ź	179	017A	pl
Ż ż	017B	017C	mt pl

Table 2. Language-specific letters of the official European Union languages using the Latin script, and Icelandic, Norwegian, and Turkish languages - Part 3.

different number of letters. There are no strict limitations on the number of keys. However customisation goes slowly and this may be due to designer fears that the keys will become too small when their number is increased.

In many existing keyboards there is a small reserve that appears in the bottom row (with spacebar) when the number of keys in letter rows is increased. The keys of punctuation marks might be moved here from letter rows and their space used for additional letters.

There are no obstacles to assigning dedicated keys for every letter of the main language in the main panel of the on-screen keyboard. The press-and-hold method is a uniform way to get many letters from every key. This method can be used to create an international keyboard with a rich set of foreign letters.

References

Bi, X., Smith, B.A., Zhai, S. (2012). Multilingual Touchscreen Keyboard Design and Optimization. *Human-Computer Interaction*, Vol. 27, Issue 4, p. 352–382.

ETSI ES 202 130 V2.1.2 (2007) Human Factors (HF); User Interfaces; Character repertoires, orderings and assignments to the 12-key telephone keypad (for European languages and other languages used in Europe). ETSI Standard. 295 p.

Grigas, G. (2014). Klaviatūros: nuo rašomųjų mašinėlių iki planšetinių kompiuterių [Keyboards: from Typewriter to Tablet Computer]. *Santalka*. 22 (1), p. 23–38 (in Lithuanian).

ISO 12199 (2000). Alphabetic ordering of multilingual terminological and lexicographical data represented in the Latin alphabet.

ISO/IEC 9995 (2009). Information technology – Keyboard layouts for text and office systems.

Letter & Word Frequencies in English / Other Language Frequencies (2014). http://www.letterfrequency.org/ [accessed on 2014-07-05].

Letter Frequency (English) - Algoritmy.net [online] (2014) available: http://en.algoritmy.net/article/40379/Letter-frequency-English [accessed 5 Aug 2014].

Lewand, R. (2000) *Cryptological Mathematics*, Mathematical Association of America. p. 199.

LST 1582 (2012). Lithuanian computer keyboard – Layout of characters. Lithuanian standard.

LVS 23-93 (1993). Latviešu tastatūra datoriem [Latvian Computer Keyboard]. Latvian standard.

MacKenzie, I. S., Tanaka-Ishii, K. (2007). Text entry using a small number of buttons. In: MacKenzie, I. S., Tanaka-Ishii, K. (Eds.), *Text entry systems: Mobility, accessibility, universality*, San Francisco, CA: Morgan Kaufmann, p. 105–121.

MacKenzie, I., S., Zhang, S., X. (1999) The Design and Evaluation of a High Performance Soft Keyboard. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems'99*, p. 25–31.

MacKenzie, I., S, Soukoreff, R., W. (2002) Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, 17, p. 147–198.

Notes

[1] http://gartner.com/newsroom/id/2674215.

Localisation Service Management Principles

Rubén Pérez García
RR Donnelley - Language Solutions
Madrid
Spain
rperezgarcia.trad@gmail.com

Abstract

Traditionally, the translation and localisation industry is considered to follow the management principles of project management. This paper presents service management principles and best practices according to the IT Infrastructure Library, currently the most comprehensive framework for service management, and discusses their application and adaptation to the translation and localisation industry. This paper identifies management practices in localisation and aligns the role of the localisation project manager to the operational functions defined in service management. It ultimately reflects on the influence of service management principles in localisation and provides grounds for further study of service management as a managerial approach for the localisation industry.

Keywords: *service management, project management, management by projects, ITIL, ITSM*

1. Introduction

Service management has evolved over the years and is nowadays a very developed field in the IT sector. The localisation industry, as a language services sector, is very much involved in service management practices. Nevertheless, it has mostly postponed the study of service management so far, or it might have circumscribed it to internal knowledge in language service providers.

There are several possible reasons for this. On the one hand, the attachment to the already well-established terminology and research focus on project management in the localisation industry. We are used to referring to translation projects and project management rather than translation service management, even if the term *language service provider* is widely used. On the other hand, it could have been the current focus on solving the issues technology presents in the use of language and management tools, which might have detracted from promoting industry talks on actual service management. Moreover, language service providers might consider they have already acquired the necessary service management expertise to meet their (and their customers') goals, through their constant reaction and adaptation to the needs of the market over the years.

At its current level of maturity, the localisation industry is ready to adopt service management best practices standards in order to improve service operations and delivery. However, there is not an established framework for localisation service management best practices yet. Thus the study of service management and its aplication to localisation can be best achieved by the analysis of existing service management standards, particularly the IT Infrastructure Library, the *de facto* standard for IT service management and the most extensive framework for service management in general.

2. From Project Management to Service Management

Localisation is evolutionary in nature. It must adapt constantly, as the software industry changes, new ways of creating content appear, and more and more content is localised. It transitioned from the unstructured efforts in the early 1980s to the creation of in-house teams with the technical and linguistic savvy to carry out localisation. In the 1990s, many language service providers emerged as the model shifted, with the IT industry often outsourcing localisation. Language service providers hired engineers, linguists and desktop publishers to handle the localisation process; more importantly, *project managers* were tasked with the coordination of complex multilingual projects (Esselink 2003).

The content explosion in the 2000s, along with the development of more sophisticated tools, standards that increased interoperability and guaranteed a certain degree of quality, as well as the consolidation of larger multi-language vendors, laid the grounds for further standardisation of the localisation workflow.

Language service providers (LSPs) gradually adapted to new market trends, increasing their ability to manage larger volumes at lower costs while incorporating value added into their services. Localisation underwent a strong transition into a modern service industry.

At present, service management (SM) clearly comes into the equation as the localisation industry attempts to find out ways to deliver content faster, with high quality and automatically, while seamlessly integrating the localisation workflow into its customers' business models. Management practices have evolved too. Even if we still retain the terminology, the very concept of *localisation project* can no longer be defined only from the traditional perspective of project management. Project management, management by projects and service management coexist nowadays in what is known as localisation project management. In order to understand the application of service management practices in the localisation industry, the relevance of each of these three separate yet related areas to localisation management must be discerned.

2.1 Project Management and Management by Projects

A Guide to the Project Management Body Of Knowledge (PMBOK) by the Project Management Institute is the main framework for project management. It gives a very concise definition of what a project is: "a temporary endeavour undertaken to create a unique product or service" (1996, p.4). Projects are initiated with a set of goals, and a clear beginning and end, and they address work that has not been done before. The PMBOK clearly separates projects and operations: operations are predictable, "on-going and repetitive" activities (*ibid*, p.4). According to the PMBOK, project management is "the application of knowledge, skills, tools and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project" (*ibid*, p.6) and it is concerned with project scope, time, cost and quality, among others.

Mantel et al make a difference between project management and general management (2011, p.5). General management deals with exceptions to the overall routine tasks of an organisation, while in projects almost everything is an exception (thus the concept of *project*). In order to accommodate projects into the workload of an organisation, restrictions concerning departmental divisions, knowledge management and budgeting, amongst others, must be lifted or adapted.

The difference between projects and operations might not always be as clear as the theory suggests, and that might be the case in localisation. Often, localisation projects are not an exception in a company's organisational framework. The production department of a localisation company might run projects according to patterns, thus streamlining many project management processes and effectively bringing them closer to operations within the organisation. Mantel et al acknowledge the limitations of the traditional definition of *project* in their well-known study on project management, *Project Management in Practice* (2011). One of the several trends that Mantel identifies in project management is in achieving routine goals. He refers to the adoption of a projectised workflow to perform routine work. Scholars such as Roland Gareis (1991) refer to this operational scenario as management by projects: the management of both projects and operations following project management practices. The PMBOK briefly acknowledges it:

> The term *project management* is sometimes used to describe an organisational approach to the management of ongoing operations. This approach, more properly called *management by projects*, treats many aspects of ongoing operations as projects in order to apply project management to them. (PMBOK 1996, p.6)

Management by projects applies to project-oriented companies, such as many LSPs, that "carry out small and large projects, internal and external projects, and unique and repetitive projects to cope with new challenges and potential in a dynamic business environment" (Gareis 1991). It could be argued that, in localisation, each project is unique. But their uniqueness in terms of *content* does not detract from the fact that translation and localisation projects can be grouped in project types or categories that follow standard or well-delimited definitions and workflows. Project categorisation enabled the development of reliable management systems, which depend on flexible yet well-defined workflows and standard practices, among others. Localisation service providers are usually project-oriented and the application of project management practices to routine work—projects with certain characteristics that fall within a project category—makes it possible to *disregard* the difference between projects and operations (or, at least, to diminish its relevance). Project categorisation is also a key enabler of the application of IT service management practices in localisation.

Steve Crago introduces some of the benefits of this approach in his white paper 'Management by Projects' (2006), most of which reflect the close connection between management by projects and service management. To name a few, he refers to the use of project management tools "fed continuously from a number of sources" (2006, p.9) as a knowledge base for organisations, to the fact that data collected from many projects can be available to different stakeholders simultaneously, and to the positive effect on customer-perceived value as all services are managed as projects, among others.

2.2 Service Management

A service is "a means of delivering value to customers by facilitating outcomes customers want to achieve, without the ownership of specific costs and risks" (ITIL 2007a, p.5). Service management is concerned with the implementation and delivery of quality services and the optimisation of the supply chain in order to do so. "Delivering value" is the key in the definition of s*ervice*, a clear goal for all service providers including localisation organisations.

Knowledge on service management is primarily centralised through the IT Infrastructure Library (ITIL), a vast framework for IT service management (ITSM), with over 1,500 pages spread throughout five core books. The large extension and the ambitious scope of ITIL are what make it the best reference, not only in IT service management, but also in general service management. Moreover, ITIL is extensive and flexible enough to allow extrapolation to sectors outside of the IT industry. Much of what ITIL has achieved, and been acknowledged for, can be applied to localisation service management in particular.

The ITIL framework was first released in 1989, with a major update in 2000 (v2). The second version soon acquired popularity, particularly the volumes on service management. The impact of ITIL resulted in the development of the ISO/IEC 20000 standard on IT Service Management in 2005. ISO/IEC 20000 comprises ITSM best practices mostly according to the ITIL framework.

The purpose of service management is to provide customers with resources and to fulfil particular needs in the form of services while meeting the customers' required levels of cost, quality and risk (ITIL 2007b, p.39). An organisation that requests a service from a specialised service provider is freed from the workload it would have to assume if it were to develop such a service on its own. Customers have their own goals which depend on the services requested in terms of ownership, control and utilisation. Service management coordinates these dependencies, for example, by providing a customer with access to its own resources as well as the service provider's outcomes and possibly to a set of resources of the latter as well. Service needs can vary greatly: a customer might only require utilisation of resources while avoiding ownership costs; or it might require ownership of resources as well.

Translation and localisation services fit this scheme, as LSPs use their resources to carry out translation projects that their customers cannot carry out on their own (for instance, because of constraints in terms of resources and expertise). LSPs also help their customers relax constraints in terms of ownership, for example, through shared ownership of a translation memory (TM), if the LSP is responsible for the maintenance of the TM while the customer retains ownership of its contents and rights of use.

ITIL refers to specialisation as one of the main principles of service management. Service providers tend to specialise as coordination of resources and tasks that are interrelated and serve a specific purpose is best placed "under the control of the group most capable" of managing it (*ibid*, p.40), and interrelated resources and tasks are grouped together so that less coordination is needed.

According to ITIL, an organisation might decide to perform activities that are outside of its core competences internally—instead of outsourcing them to a service provider with the expertise to carry them out—*if* such an organisation is *confident* on its capability to do so and deems the project *feasible*. This might sometimes be the case in the localisation industry, where the large scale of some multilingual projects drives some corporations to create their own localisation departments so as to better coordinate outsourcing efforts, or to carry out localisation projects internally, or even to apply some of the most recent localisation models that are gathering momentum, such as crowdsourcing.

3. IT Service Management Functions and Localisation Project Management Roles

Achieving a consensus on what localisation project management comprises is a challenge, as the responsibilities of a project manager in one LSP might be significantly different to what project managers (PMs) do in other LSPs. A better

understanding of localisation project management can be acquired through the analysis of the actual work performed by project managers in different localisation companies. DePalma and Pielmeier's report for Common Sense Advisory, *The Responsibilities of Project Managers* (2013), is a useful source to find out which tasks PMs often perform.

Based on a survey of 409 localisation project managers representing 182 LSPs from 52 countries, the report reveals that, out of the 44 tasks listed in it, none of them are performed by all respondents. There are, however, certain patterns which might serve as an indication of what localisation PMs often do. Their most common responsibilities are related to the traditional translation and localisation workflow, and include project planning and execution, some financial aspects (quoting), communications management, file management, and signing off projects for delivery (*ibid*, p.5). Other, peripheral functions include recruiting vendors, formatting tasks and training other employees, and many PMs also have secondary roles as linguists or account managers. Multitasking is frequent among PMs.

According to the survey, localisation PMs "handle the vast majority of production and client-related issues" (*ibid*, p.6) and serve as the contact to the customer in most cases (73%). This can be linked to the strong customer focus in service management best practices, which stress the importance of efficient incident resolution.

The two factors that make the most difference in respondents' answers are experience and rank and whether PMs work in general or specialised projects. Different profiles of PM perform different tasks, pointing out access management, hierarchy and escalation as well as task assignment—all of them considered in ITSM—as important managerial aspects in localisation organisations.
DePalma and Pielmeier describe the importance of support positions to PM functions as "critical" (*ibid*, p.16). They hint at the intermingled relationship between projects and operations in localisation companies, as PMs are supported by assistants, functional production teams and "even operational teams" (*ibid*, p.16).

Unlike management by projects or agile project management, IT service management is not an evolution on management practices based on traditional project management, but a methodology in its own right, based on the Deming cycle (plan-do-check-act). Therefore, even if much of the terminology and many processes in project management and ITSM are shared, there is not a specific project management role in ITIL. In fact, ITIL does not usually refer to projects, but to services. The terms *project* and *service* are neither mutually inclusive nor exclusive. Services can be provided through both projects and operational activities; at the same time, projects can be carried out for both services and products.

According to *Service Operation* (ITIL 2007d), there are four main functions in service operations management: service desk; technical management; IT operations management; and application management. Each of these functions is carried out by a team, a functional unit within the organisation, in which different roles must be fulfilled (ITIL lists over 20 different roles spread over them). As pointed out above, a localisation PM is supported by different functional units in its organisation, effectively acting as a cross-functional link among stakeholders. For this reason, circumscribing the role of localisation project managers to a single function in ITIL is not possible. Nevertheless, some functions are closely related to the most common responsibilities of localisation PMs: service desk and IT operations management. Some technical and application management activities can be linked to the role of a localisation PM on occasion, yet these functions are more directly linked to what the leads of particular departments in an LSP would do.

The definition of the service desk already brings to mind some of the responsibilities of localisation PMs:

> The Service Desk . . . should be the single point of contact for IT users on a day-by-day basis—and will handle all incidents and service requests, usually using specialist software tools to log and manage all such events. (ITIL 2007d, p.198)

The service desk is the point of contact for customers, which helps improve and clarify accessibility to information. With an effective centralised service desk, customers are likely to receive answers to their requests faster, and requests are sent to the relevant stakeholders and solved in a controlled manner. This improves both communications and information management.

Service desks in ITSM are responsible for logging and escalating incidents, informing customers and

receiving their feedback, and updating the information in the management system accordingly (2007d, p.199). ITIL also refers to the possibility of creating specialist groups in the service desk to handle particular services, much like what specialist PMs do according to CSA's *The Responsibilities of Project Managers*.

As for particular roles within the service desk function, *Service Operation* includes the service desk manager, who reports to senior management; the service desk supervisor, who acts as the escalation point for complex issues; and the service desk analyst, who handles service requests, reports incidents, performs request fulfilment and provides first-level support.

Operations management refers to "the department, team or people responsible for performing the organisation's day-to-day operational activities" (*ibid*, p.227). The IT operations management function encompasses the execution and monitoring of activities, as well as tasks such as job and shift scheduling and transitioning plans into actions. IT operations require an efficient use of resources to save costs and focus on achieving a return on investment strategy. Important information about activities in IT operations must be logged (e.g. completion of jobs, delivery, performance, and so on).

As for particular roles within IT operations management, *Service Operation* includes the IT operations manager, in charge of monitoring operational activities and scheduling routine work; and the shift leader, who takes responsibility for decision-making and control of activities.

The cross-functional nature of the localisation PM position implies that some of the responsibilities of other roles listed in *Service Operation* are related to what a localisation PM does. These include the incident manager, who handles incidents and the information regarding such incidents; and the problem manager, who performs follow-ups on major problems, acts as the contact point for suppliers and makes sure they fulfil their obligations.

4. ITSM Principles in Localisation Service Management

Senior management is responsible for *deciding* company policy regarding the market needs an organisation intends to address, the workflows and tools used, the organisation's strategic goals, and so on. Project management is one of the most—if not the most—relevant function in *executing* company policy. The importance of project managers in executing company policy should be considered not only regarding *what* project managers do but also *how* they do it.

There are two main factors that impact the work of localisation project managers: the management principles in place and the management system in use. Technology considerations are essential in this regard: it is hardly possible that project managers will be able to efficiently follow the management principles of a localisation organisation if the management system itself—usually a translation management system (TMS) in LSPs—does not enable them to do so. For over a decade, LSPs have embraced advanced and constantly evolving management systems, in an increasingly competitive environment, with the goal of reducing overhead and increasing efficiency and throughput. Currently, many LSPs rely on complex translation management systems to manage most operational aspects. To a large extent, TMSes are intended to capture and spread company policy in an organisation. Sargent and DePalma (*Translation Management System Scorecards,* 2007) outline three main TMS categories: translation-centric solutions, business management solutions and enterprise solutions. This concurs with the idea that there are various business and production models among LSPs. A TMS that conveys the principles of service management is likely to be a great asset for organisations that aspire to instil SM best practices into their operational model.

The service lifecycle (see figure 1) is structured around five main stages in ITIL: Service Strategy, Service Design, Service Operation, Service Transition and Continual Service Improvement. This paper mainly draws upon the first three, as they provide an overview on service management that accounts for the principles of service management.

Service Strategy (ITIL 2007b) is at the core of the service lifecycle. It focuses on the role of service management towards meeting business goals, the relationships between customers and service providers, service provider types and the organisational aspects of service management. *Service Design* (ITIL 2007c) is concerned with the practical aspects in the application of a service strategy, in terms of *utility* and *warranty*. *Service Operation* (ITIL 2007d) focuses on the processes and

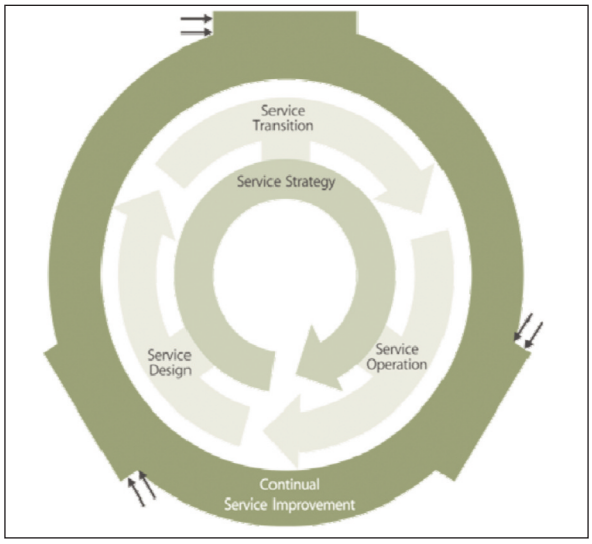

Figure 1: Service lifecycle (ITIL 2007b, p.45).

functions in service management necessary to successfully operate services.

4.1 Customer-Perceived Value of Services

Service customers have two main expectations. On the one hand, a high degree of *utility*, that is to say, an increase in the performance of their assets. On the other hand, *warranty* that a potential deviation in service performance will not offset the benefits in utility.

The concepts of service warranty and utility are related to managing uncertainty. Customers need to be reassured that, at a certain cost, they will obtain a service of a certain quality. Warranty backs up utility and shifts the costumers' concerns from the risks and doubts of demanding a service to the potential gains the service yields.

4.1.1 Utility

Utility means "fitness for purpose" (ITIL 2007b, p.54). It is communicated by means of establishing clear outcomes for a definite service from the beginning of the service lifecycle, as well as in terms of the ownership costs and risks the customer avoids.

Commonly, service providers help remove or relax certain constraints their customers find in their business strategy. For example, the translation of a website removes a constraint in access from the people of a locale. This is perceived by the customer as a gain. Moreover, utility also comes from the risks avoided by the customer by purchasing a service from a specialised service provider instead of developing and carrying out certain tasks on its own. Typical constraints a customer might find are a limited capacity to perform some type of work internally or maintaining non-core assets to perform sporadic work (with the associated costs this approach implies).

Regarding the **outcomes supported** by the service provider, the concepts of *service portfolio* and *service catalogue* are relevant here. The service catalogue of an organisation includes all services the organisation markets, offers and, consequently, is capable of delivering. In localisation, these services might be website localisation, software localisation, legal translation, interpreting and so on. The service portfolio covers a wider range of services. It includes all services an organisation is capable of delivering even though they might not be part of their core offering and the organisation does not actively market them. The service portfolio might include third-party services, some of which might not be visible to the customer. These services are most often supporting services required to perform the core services (e.g. in localisation, desktop publishing, graphic design and machine translation, notwithstanding whether these services are outsourced or insourced). Moreover, the service portfolio includes services that the organisation no longer offers or carries out, that is to say, *retired services* (*ibid*, p.120).

The inclusion or not of a service in the service catalogue is not dependent on the nature of the service itself but rather on service strategy. That is to say, the above examples are just likely candidates and

not set in stone. Desktop publishing, for instance, might be included in the service catalogue of an LSP if it fits company policy regarding availability, capacity, value creation and market interest, among other factors.
Flexible service portfolio management involves a high degree of customisation that, for each new service, enables specific key performance indicators and metrics, financial reporting, status information, integration with third party tools, customisation of workflow designs and responsibility assignment, and individual definitions to be created for each service. In localisation, the TMS should ideally enable this degree of parametrisation and functionality.

A key reason that drives customers to require a service from a specialised service provider is that of **avoiding ownership costs and risks**. An organisation that requires a service outside of their core competences and tries to obtain it internally often finds itself constrained by the lack of expertise and, most likely, by the inability to estimate the associated costs and risks accurately. The demand of a service from a service provider not only helps the organisation obtain result-oriented benefits. It also helps remove such constraints and derives accountability to the service provider, at least to a certain extent. This is a clear motivation for customers of localisation services.

An organisation that outsources translation of, for instance, a legal or medical text to an LSP does so, on the one hand, to avoid the risk of incidents (lawsuits, negligence) from happening and, on the other hand, to derive ownership of such risks to the LSP (through a contract, a service-level agreement or other means), among other reasons. Accountability is transferred to the service provider or shared with it, and this creates utility, that is to say, it is perceived as added value by the customer. Undoubtedly, the outcomes (time, cost, performance) from the translation service are extremely important, but the ownership costs and risks avoided are a benefit not to be neglected.

4.1.2 Warranty

Warranty means "fitness for use" (ITIL 2007b, p.35). Utility and warranty are interconnected: a service that meets its required purpose might still be a failure if it is not delivered *how* the customer needs it in terms of availability, capacity, continuity and security, defined in *Service Design* (ITIL, 2007c) as the core processes of correct service design.

According to ITIL, a higher level of warranty is what makes some service providers stand out from the crowd, as warranty offers competitive advantage. ITIL refers to this circumstance in a way that echoes very closely changes currently taking place in the localisation industry, as the value of warranty as a basis for competitive advantage "is particularly true where services are commoditized or standardized" (ITIL 2007b, p.61). In a commoditised industry, the value of utility becomes harder to distinguish among service providers; therefore warranty becomes a strong selling point. Commoditisation leads customers to assume a certain quality and to start focusing on *how* they are delivered a service (faster, safer, cheaper, and so on).

There are several ways to communicate warranty of provided services to customers. ITIL points out certainty and transparency as two of them. A customer might need to know in detail the work that needs to be carried out before he is delivered a particular service, so as to know certain, predictable conditions of the delivery of a service. One of the effects of the application of quality standards such as the ISO 9001 and the EN 15038 standards is adding predictability to translation and localisation services.

Capacity is concerned with matching resource availability to business needs in order to maximise resource efficiency while reducing response times and the risks associated with unexpected workload increases (ITIL 2007c, p.134). Capacity directly affects service warranty. Service providers must be prepared to deal with a non-constant work stream and peak demand. This type of flexibility ensures that a service will be available in a timely manner even if there are changes in requirements and/or inputs from the customer. It also increases reliability, as periods of peak demand serve as a test where the service provider can show the opportunity gains from their service offering. Capacity management is also concerned with identifying patterns (and volume) of service requests so as to better adjust performance requirements and help service-level management better understand the customers' capacity requirements.

Capacity constraints in the industry are not usually linked to server speed or storage limits, but to human resources in the supply chain. Most LSPs create and maintain a detailed vendor database which includes suppliers (both companies and freelancers) covering all services in their service portfolio, from translators and reviewers to DTP specialists and IT companies.

Availability is one of the most visible aspects of

service warranty from the customers' perspective: a service should be delivered on time and, at delivery, it should be readily available for use under the agreed-upon conditions.
The high visibility of availability as value added makes for good availability management, a competitive advantage in the localisation industry (and most service industries). It should be regarded as a complex management field that goes far beyond the promise of shorter delivery times. Different levels of availability are vital so as to offer service warranty. Depending on the potential deviations in service management, incomplete instances of a service deliverable might be made available for use to a customer (or other particular stakeholders) so delays will still leave a customer some room to manoeuvre.

Commonly, customers demand confirmation of availability as soon as possible after they place a service request. If the right metrics are in place, delivery times might be estimated accurately. Availability of a service can only be high as long as it is reliable. A system offers reliability as long as it provides comprehensive, relevant information to help the manager determine availability. Availability issues in a particular component might have a ripple effect and ultimately affect service delivery, therefore solving them should be a priority.

Continuity is related to risk management in the wider sense. For the most part in ITIL, continuity is centred around IT infrastructure and short-term risks from critical failures in IT components (service disruption). In the context of localisation, we regard continuity in relation to the possibility of recovering past work or backtracking to previous processes in case of failure or error.

Continuity is even more important in localisation than in many other service industries. As an industry heavily reliant on leveraging past work, continuity failures could potentially have disastrous consequences for an LSP. The customers' business decisions are impacted by post-service support, and leveraging content (terminology, translated segments, and so on) is a great asset for any LSP that wants to remain competitive. Due to the costly implications of continuity issues, LSPs tend to adopt the rough measure of creating server-based back-ups of absolutely all contents and data handled and installing recovery systems to mitigate any potential loss. Even if it can be argued that this reactive measure is not disproportionate in the context of localisation, full back-ups are not intended for any and all continuity failures. Cost-efficient continuity management also requires more compartmentalised reactive measures, and proactive measures must be considered as well.

Risk analysis involves inputs from availability, security and capacity but it is mostly considered integral to continuity. An established methodology, such as Management of Risk (ISO 31000) is recommended in ITIL. Management of Risk follows the cycle of identifying and assessing risks, planning, then implementing solutions, thus reducing the chance of potential risks.

Information security implies avoiding risks by using a customer's assets only if authorised and for the agreed purposes. This involves providing access to the relevant stakeholders only and under the customers' approval, and protecting the customers' assets from "unauthorized or malicious access" (ITIL 2007b, p.60).

Information security is paramount in localisation as service providers retain and maintain their customers' assets in the form of files and documents, and store information in translation memories and terminology databases (TDB), among others. Notwithstanding to whom the property of the content within TMs and TDBs, databases and documents belongs (whether to the customer or the service provider), it is in the best interest of an organisation to maintain a tight policy on information security.

Information security management in ITIL is structured around three principles: availability, confidentiality and integrity. Information availability involves providing access to information to the relevant stakeholders. Moreover, information security must ensure that integrity is kept, that information remains complete and cannot be modified without authorisation. Regarding confidentiality, security management is responsible for avoiding information leaks and unauthorised access. As an example, it is common practice in the localisation industry to ensure confidentiality by means of non-disclosure or operational-level agreements signed by vendors.

4.2 Service Operation Processes and Management Activities

There are three main aspects to service operations: processes, activities and functions. The third refers to the roles played by different member of an organisation in the service lifecycle and have been briefly referred to above. The focus on processes and

activities implies that this area is more concerned with particular features that enable the application of service management best practices through a management system, commonly a TMS. Particularly, management activities in ITIL refers to the use of technology features that enable and ensure that management principles are followed according to service strategy and company policy.

4.2.1 Service Operation Processes

In ITSM there are five main processes for appropriately carrying out and monitoring service operations: event management, request fulfilment, access management, incident management and problem management. *Service Operation* (ITIL v3, 2007d) covers these processes in depth. The correct implementation of service operation processes supports competitiveness and centralises knowledge on operations performance. ITIL operation processes are very relatable to service management in general and localisation service management in particular. Adequately managing project requests, monitoring and escalating issues appropriately, delivering outcomes to the relevant stakeholders and preventing potential disruptions to operations are some of the core ideas behind ITIL service operation processes. All of these aspects are most likely a top priority for most localisation organisations.

According to ITIL, an event is "any detectable or discernible occurrence" with an impact in service delivery or in the management of the IT infrastructure (2007d, p.67). Events can originate from either regular operations or deviations from standard service operations. Event management is used to measure actual performance against expected performance of a service, as well as to detect potential incidents early.

Event management implies automation of certain monitoring events so as to help managers carry out their work efficiently. In an LSP, this usually takes place by means of particular functionality in the TMS in use. Event management requires passive monitoring of actions and, above all, alerting the user about events. Notifications of all sorts often signal events: a translator has uploaded a file, a user has logged into the system, new e-mail has arrived, a delivery date for a project has not been set, a file could not be delivered or uploaded to the system, a task has not been assigned, costs of a project exceed its assigned budget, and so on. The list is almost endless. It should be noted that, even if some events might point out potential issues (e.g. a file could not be delivered), they should not be considered incidents as long as they do not require diagnosis and intervention; in other words, most events can be dealt with immediately and notifications are there to provide the user with meaningful information or exhort them to take remedial action immediately. If that is not the case, events should be reported as incidents.

An incident is an "unplanned interruption" to a service or a "reduction in the quality" of a service (2007d, p.86). Issues that have not yet impacted service operations but might do so in later stages are also considered incidents according to ITIL. Incidents should be dealt with in order to resume service operations or to avoid interruptions. A problem, on the other hand, is "the cause of one or more incidents" (*ibid*, p.111).

Incident and problem management are concerned with detecting, diagnosing and resolving such occurrences. Typical incidents include system failures and—commonly in localisation—queries sent by stakeholders. In localisation, PMs are usually responsible for passing queries sent by linguists to the relevant stakeholder (e.g. the customer's validator). It is not uncommon that they cannot resume their work until the queries are resolved. Technical issues are also common incidents. Usually, workarounds are used until a problem is diagnosed and resolved, so as to minimise its impact on current operations.

The goals of **request fulfilment** include providing "a channel for users to request and receive standard services" (ITIL 2007d, p.105), informing about service availability and pricing, delivering outcomes and gathering feedback. Service requests do not only involve services themselves but anything related to services (e.g. information and quotes). Requests differ from incidents in that—whereas incidents are unplanned—they are planned or expected (*ibid*, p.105), therefore an organisation must be prepared to fulfil requests in a predictable way. It is within the policy of an organisation to decide how it will handle service requests. This includes quoting as well as initiating projects and, on the other hand, delivering outcomes and terminating projects.

Access management in ITIL *Service Operation* is quite focused on IT, but can be extrapolated to general service management. Access management is directly related to security management, but refers more specifically to granting access to a service to the relevant people, whereas security management was also concerned with the strategy of an

organisation in keeping confidentiality and content integrity. Access management is, nevertheless, the application of both security and availability management principles when running service operations. It should be noted that a user who manages access rights (e.g. a PM) is not responsible for deciding who has access to which information or content. That is within the scope of security management. Access management is responsible for the execution of the access policy established by the organisation.

Access management is extremely important for localisation organisations and for service providers in general. Controlling access to information and assets serves various purposes. Granting specific access rights enable stakeholders to carry out their assigned tasks while preventing errors and avoiding undesired modifications of information and/or assets by unauthorised users. And, undoubtedly, it is necessary so as to keep confidentiality. This management process is also known as identity management, as it also involves saving the appropriate information on users and verifying their statuses.

4.2.2 Activities

According to ITIL, activities "ensure that technology is aligned with the overall Service and Process objectives" (2007d, p.146). Activities support service operations and mostly refer to technology management. Admittedly, ITIL does not provide a list of activities for general service management. This is due to the fact that different service sectors might require different technology or different approaches to use similar technology.

For a mature service organisation, technology is a means to achieving business goals. This is easier said than done. For this to be true in the case of the management system in use in an organisation, there must be a correlation between the organisational approach to technology management and the capabilities of the management system itself. This is the idea behind service operation activities in ITIL, which makes it necessary to consider localisation-specific activities in the application of service management to localisation. Technology management activities should above all align service goals and technology tools.

Service Operation (ITIL, p.2007d) includes a number of **service operation activities** required to manage IT. Even though many of them are very specific to ITSM, some can be applied to service management in general and localisation service management in particular. These include, for instance, naming conventions and folder structures, interconnectivity, communication among users and re-routing of workloads, monitoring and control of key operational tasks, the implementation of a well-designed operations bridge, systems support, database administration and automation of repetitive tasks, among others.

As for **localisation activities**, some of the most common among them can be identified in the evaluation of TMSes carried out by Benjamin B. Sargent and Donald A. DePalma in their CSA reports *How to Select a Translation Management System* (2011), *Translation Management System Scorecards* (2007) and *Translation Management Systems* (2008). Among others, support for translation tools, quality standards, file management and workflow management can be discerned as some of the core technology activities relevant for LSPs and with a key role in aligning the functionality of a management system with the business and operational needs of LSPs.

5. Conclusions

At present, a detailed framework for localisation service management does not formally exist. Nonetheless, ITIL provides the necessary knowledge to understand service management and enables the identification of key ideas in service management that play a role in localisation. Even if project management accounts for many management aspects in localisation organisations, it can hardly continue to be considered the main and only management framework in an industry that is increasingly reliant on technology, focusing on the goal of meeting customers' needs, competing in a commoditised market where competitive advantage is acquired by means of improving *how* services are delivered rather than *what* is delivered. Quality alone is not enough. Efficiency is a driving force as well. Service management is much more inclusive of technology considerations than project management—partly due to its strong development in the IT industry—and it still places the necessary importance on the people and processes involved in service operations and delivery. Efficient service management, with its clear focus on reducing overhead and improving processes, offers key benefits to localisation organisations.

LSPs that have achieved a high degree of maturity often have project managers specialised in particular

fields. Even if the roles defined in ITIL cannot be directly matched to localisation project managers, these usually have responsibilities which are similar to the responsibilities listed in ITIL *Service Operation* for several of the most common roles in service management.

Service management can be closely linked to localisation management. It is likely that experienced localisation project managers would recognise the importance of SM principles such as availability and capacity management, access management, monitoring or request fulfilment in management practices in localisation organisations. Further research on service management and the incorporation of industry talks into the study of service management would move forward the discussion and help communicate the idea—frequently expressed in this paper—that the establishment of a standard framework for localisation service management would help the localisation industry improve efficiency of project management and operational activities, even if the challenge in doing so is considerable.

References

Choudhury, R. and McConnell, B. (2013) *Translation Technology Landscape Report*, De Rijp: TAUS DV.

Crago, S. (2006) *Management By Projects* [online], Vital Business Solutions Pty Ltd, available: http://www.alertit.com/Whitepapers/Alert%20Management%20By%20Projects.pdf [accessed 25 October 2014].

Dunne, K. J. and Dunne E. S., eds. (2011) *Translation and Localization Project Management: The art of the possible*, Amsterdam: John Benjamins Publishing Company.

Dunne, K. J., ed. (2006) *Perspectives on Localization*, Amsterdam: John Benjamins Publishing Company.

Esselink, B. (2003) 'The Evolution of Localization' [online] in Multilingual Computing and Technology. *Guide to Localization*, 4-7, available: http://isg.urv.es/seminars/2003_localization_online/esselink.pdf [accessed 25 October 2014].

http://www.multilingual.com/downloads/screenSupp57.pdf [accessed 25 October 2013].

Gareis, R. (1991) 'Management by projects: the management strategy of the "new" project-oriented company' [abstract]. International Journal of Project Management, 9(2), 71-76.

Mantel, S. J., Meredith, J. R., Shaffer S. M. and Sutton M. M. (2011) *Project Management in Practice,* Hoboken: John Wiley & Sons.

OGC (2007a) *The Official Introduction to the IT Service Lifecycle*, London: TSO.

OGC (2007b) *ITIL Version 3: Service Strategy*, London: TSO.

OGC (2007c) *ITIL Version 3: Service Design*, London: TSO.

OGC (2007d) *ITIL Version 3: Service Transition*, London: TSO.

OGC (2007e) *ITIL Version 3: Service Operation*, London: TSO.

OGC (2007f) *ITIL Version 3: Service Improvement*, London: TSO.

Pielmeier, H. and DePalma, D. A. (2013) *The Responsibilities of Project Managers*, Lowell: Common Sense Advisory.

Project Management Institute (1996) *A Guide to the Project Management Body Of Knowledge*, Newtown Square: Project Management Institute Publishing Division.

Sargent, B. B. and DePalma, D. A. (2007) *Translation Management System Scorecards*, Lowell: Common Sense Advisory.

Sargent, B. B. and DePalma, D. A. (2008) *Translation Management Systems*, Lowell: Common Sense Advisory.

Sargent, B. B. and DePalma, D. A. (2011) *How to Select a Translation Management System*, Lowell: Common Sense Advisory.

Wittner, J (2004) 'Strategic Knowledge Management for the Localisation Industry", *Localization Reader*, 2004-2005, 69-72. [From *Multilingual Computing & Technology*, 15(4)].

Upgrading Mobile Applications Dynamically through Crowdsourcing for Including New Languages

Taygun Bulut Durmaz, Susana Muñoz Hernandez
Departmento de Lenguajes y Sistemas Informáticos e Ingeniería de Software,
Facultad de Informática, Universidad Politécnica de Madrid, SPAIN
tb.durmaz@alumnos.upm.es, susana@fi.upm.es

Abstract

There are multiple workshops and markets available on the web where communities of programmers develop mobile applications for several purposes. Most of these applications are not restricted to a particular country or region. In fact, the users of a mobile application might speak many different languages. According to research, more than 99 percent of mobile applications are developed in 9 languages. However, there are more than 6000 languages spoken around the world. For those people who do not speak one of these 9 languages, how are they supposed to use most mobile applications on the market? The MilanApps project has an answer for this problem. Mobile applications usually have user interfaces that contain text in the form of words or sentences. In order to make the application usable by people who speak different languages, this text would need to be translated. The MilanApps project will make this feasible by providing translations of the user interface text of mobile applications to any language.

Keywords: *Localisation, translation, mobile apps, TEDECO, MilanApps*

1 Introduction

1.1 Mobile Applications

A mobile application, or mobile *app*, is a computer program designed to run on smartphones, tablet computers and other mobile devices.

Mobile apps were originally offered for general productivity and information retrieval, including e-mail, calendar, contacts, stock market and weather information. However, public demand and the availability of developer tools resulted in a rapid expansion into other categories, such as games, factory automation, GPS and location-based services, banking, order-tracking, ticket purchases and recently, medical apps. The explosion in the number and the variety of available apps made new app discovery a challenge, which in turn led to the creation of a wide range of sources for review, recommendation, and curation, including blogs, magazines, and dedicated online app-discovery services.

The popularity of mobile apps has continued to rise, as their usage has become increasingly prevalent across mobile phone users (Ludwig 2012). A comScore study in May, 2012 reported that in the previous quarter, more people used their mobile devices for apps than for browsing the web: 51.1% vs. 49.8% respectively (Perez 2012). Researchers found that usage of mobile apps strongly correlates with user context and depends on user's location and time of the day (Böhmer *et al.* 2011).

1.1.1 Demands

We estimate that native English speakers account for only 34%, 39%, and 25% of iOS, Android, and Windows Phone downloads, respectively. (See Figure 1)

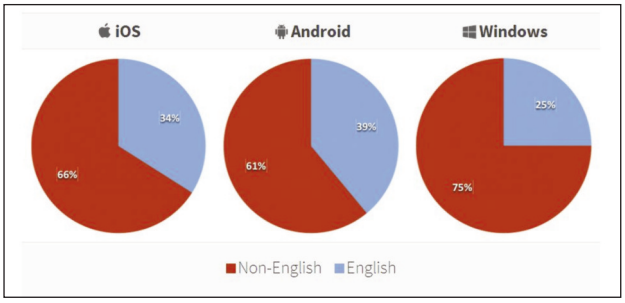

Figure 1: Market demands according to majority platforms from Xyologic (Tethras 2013)

iOS		Android		Windows	
ZH Chinese (Si.)	38.8%	KO Korean	18.0%	ZH Chinese (Si.)	27.8%
JP Japanese	13.5%	ES Spanish	17.8%	ES Spanish	13.9%
ES Spanish	8.4%	RU Russian	10.9%	IT Italian	10.7%
DE German	6.4%	DE German	9.5%	RU Russian	10.0%
KO Korean	5.7%	JP Japanese	8.0%	PT-BR Portuguese (Br.)	6.1%
FR French	5.7%	FR French	5.3%	DE German	5.7%
RU Russian	5.3%	PT-BR Portuguese (Br.)	5.2%	FR French	4.8%
IT Italian	4.3%	IT Italian	4.8%	PL Polish	3.8%

Figure 2: Market demands according to majority platforms from Xyologic (Tethras 2013)

Native English speakers currently make up approximately 33% of the total addressable market on average. If we consider the non-English market among the three major platforms (iOS, Android, and Windows) by concentrating on the top 8 languages, we have the following result (See Figure 2).

The results show that the English market has 33% (average) of total downloads, followed closely by the Chinese market with 23%, and then the Spanish market with 13%.

The situation is unlikely to change as current demand determines future supply. In other words, developers and companies attempt to meet their customers' expectations according to their demand. This situation tends to maintain English, Chinese, and Spanish as the main languages of mobile apps.

1.1.2 Results

The market chain of mobile apps causes an imbalance between spoken languages versus app languages since 85% of developers publish in English addressing just 8% of the world population (around 500 million people) who speak English as a primary language. Meanwhile, Chinese, spoken by 22% of the world's population, only attracts 16% of developers. The choice of English by the majority of developers happens not only on a global, but on a regional basis as well, putting local languages supply at a deficit.. In South America, Spanish is used by 84% of developers, while English is only used by 48% (Kapetanakis 2012).

1.2 Minority Languages

According to UNESCO 6,000-7,000 languages are spoken in the world today. However, about 300 widespread or majority languages account for 90 percent of the world's population. More than half of the remaining languages are endangered.

"Minority languages have been oppressed, denied, and neglected for a long time, and decline is accelerating. Whereas estimates show that half of the world's languages disappeared from 1450 to 1950, half of the remaining 6000 to 7000 languages could disappear in this century alone" (Sheyholislami 2009).

Language plays an important role in learning. Since language is the main medium of communicating meaning in most learning activities, it is essential to use in education a language that learners understand and speak. Usually people understand their first language best, and are most comfortable speaking it. Multilingual people may be equally proficient in several languages. The first language is also often called the mother tongue, or the home language. Generally, the first language is a language one has learned first; one identifies with or is identified as a native speaker of by others; or one knows best (Unesco 2005).

Some observers consider new media technologies such as satellite, television and the Internet, as the salvation of minority languages. They believe it is important to maintain and develop minority languages and see the media as having a crucial role in maintaining and developing minority languages. Minority language media are deemed important for:

- their symbolic role in acknowledging that minority cultures can deal with the contemporary world;
- their ability to legitimate the existence of the language that they use;

- their potential to provide an "economic boost" for those who are interested in working in the minority language;
- their instrumentality in engendering a public sphere within a language community;
- their resourcefulness in enabling minorities to represent their community, not only within itself but also to outsiders instead of being represented by "others";
- their capability to be conveyers of cultures and producers of cultural products; and
- their capability to magnify discursive practices of identity construction (Cormack and Hourigan 2007).

1.3 Proposal

There are multiple forges and markets available on the web where communities of programmers develop mobile applications for several purposes. Most of these applications are not restricted to a particular country or region. In fact, the users of a mobile application might speak many different languages. The most common scenario is that software is only available in one or two popular languages, such as English, French, Spanish or Chinese. Another possibility is that the software is available only in the language of the developers.

But what is the consequence of not having software adapted to local languages? That the software is in many cases restricted to people who know foreign languages. What about applications for the common population? What about applications for children?

The consequences for software applications, that try to be popular in the market, are dramatic because its target market is limited by the language of its user interface messages and tags.

It is often the case that the developers are the ones writing and translating the user interface text of mobile applications. But why not open the work to the collaboration of non-technical staff in the development team? They could aid in translating the user interface text of software applications to local languages.
The idea behind this proposal is, as in other issues where user collaboration is assumed (for fulfilling opinion surveys, to evaluate efficiency of services, to provide feedback, etc), to take into account the collaboration of users to load new languages to mobile applications.

The effective way of implementing this is not technically difficult but implies logistic planning, including a review process and an automatic way of loading new languages, which is currently not a global recommendation though it probably should be one. Here we provide a proposal for doing this in a simple and automatic way that would work for any software project intended to be multilingual.

2 State of the Art

2.1 MilanDi Project

The MilanDi project can be considered a "baby step" towards one of our aims: saving the minority languages. MilanDi (Multilingual dictionary for minority languages) is a dictionary for the minority languages that are spoken around the world. The goal was to create a mobile application using Universal Natural Language (UNL) and develop it together with other languages.

This open source project is composed of two main parts. The first part is a webpage where people around the world can contribute with translations to their own languages and by doing this, enrich the MilanDi system database. The second part is a mobile application that serves as a client for the MilanDi system in Android mobile devices.

The MilanDi webpage contains around 6000 UNL words, currently including English, French, Spanish, Swahili, and Kirundi. Translators can login to the system and translate words and sample sentences to their own language. Thanks to the contribution of local translators, the MilanDi project expects these 6000 words to be translated to many languages, especially minority languages. The MilanDi webpage will be published on TEDECO servers after acquiring the necessary permissions.

The MilanDi mobile application is an Android application that uses the collected data on the webpage. Users can use the application to translate words according to their needs. They can also update the word database through the application. If the mobile phone is connected to the Internet, the application fetches the new database from the MilanDi webpage and updates the offline database of the application.

MilanDi is a successful project that expresses the power of collaborative contribution. Instead of searching for the translation of a word, which is almost impossible for some minority languages as they are not available on the Internet, local translators

can contribute to this dictionary to save their own language.

3 Prototype: MilanApps

3.1 Description

The MilanDi project made us realize that collaborative contribution could be a very powerful tool in preserving and even spreading minority languages. We then decided to get contributions from local translators for language support of mobile applications.

Mobile applications usually have user interfaces that contain text in the form of words or sentences. In order to make the application usable by people who speak different languages, this text would need to be translated. The MilanApps project will make this feasible by providing translations of the user interface text of software applications to any language.

The MilanApps project has two main aims in its essence. The first aim is for people living in developing countries to use mobile applications in their own language. The second aim is to save minority languages through increasing awareness of them.

In order to accomplish these aims, we have created a portal for developers to work in collaboration with reliable translators. Developers of mobile applications will upload their user interface text to the MilanApps portal and reliable translators will translate them into their language of expertise. Therefore, developers will have the capacity to create multilingual applications for mobile apps users.

People understand their first language best, are most comfortable speaking it, and for this purpose, learning activities should mainly be done in their own language. The MilanApps project gives them this opportunity. People will be able to use software in their own languages. Consequently, people will be able to cope with new technologies, at least mobile applications, much more easily.

Additionally, MilanApps will help to spread minority languages in mobile applications. According to Portio Research, there were 1.2 billion mobile application users worldwide in 2012 and this number was expected to reach 4.4 billion users at the end of 2017 (Portio Research 2013). If minority languages can enter a market with such vast potential as that of mobile apps, the awareness of them in the world would surely increase.

So how will this system work?

1. Developers around the world will send their application name with the text they have used on their user interface.
2. After their request is approved, the text will be recorded on the MilanApps system and will be visible to translators around the world.
3. Reliable translators will translate the text to their language of expertise.
4. Finally, developers will fetch the translated text.

We have also created an Android-based test application for MilanApps to translate its text into a variety of languages. Reliable translators have connected to our local server and contributed to the project by translating the UI text of this test application to their local languages. We received the results in one day and our test application currently supports English, Turkish, Spanish, and Kurdish. You can see the screenshots in the "Translation Forms" section.

The MilanApps project prototype is ready and working appropriately. It will be published on TEDECO servers after acquiring the necessary permissions.

Our expectation for this project is for both developers and local translators to contribute to the system. With the help of more people, more applications can be translated into local languages and our project will achieve its goals. If this project is able to spread around the world and become a standard, people would be capable of using mobile applications in their own languages. As a result, language barriers would decrease and minority language speaking populations could reach technology much easier.

3.2 Open Source and Free to Use

The MilanApps project is developed for the people who do not have access to technology in their own languages. We want to increase language support for mobile applications not only for majority languages, but also for minority ones.

It is possible to find similar platforms to MilanApps on the Internet. They intend to provide a solution for the localisation of mobile applications. However, one

cannot solve a minority language problem without knowing the expectations of minority language speaking users. These platforms only support majority languages or request money for translations. This situation severely hampers localisation. First of all, it is useless for minority languages if the platform only supports translation of majority languages. Secondly, developers are less likely to give language support for minority languages if they need to pay for the translations.

The idea behind the MilanApps project is that everyone can contribute with translations to help other people. Since this is a free-to-use platform, both developers and translators can contribute without having any extra expenses. Developers can upload their UI text to the platform and wait for translators to translate those into local languages. Therefore, developers can easily add extra language support without much effort.

Moreover, this project is Open Source so any developer around the world can contribute to improve the platform according to user needs. As a result, MilanApps gets its power from being Open Source and free to use. Collective collaboration has an amazing potential for translations. MilanApps taps into this potential as part of a non-profit organization, TEDECO.

3.3 SourceForge Repository

SourceForge is a web-based source code repository. It acts as a centralized location for software developers to control and manage free and open source software development. It was the first platform to offer this service for free to open source projects (James 2007).

MilanApps has a repository on SourceForge where developers around the world can contribute to our prototype. This promising open source project hopefully will help to universalize the translation of the mobile applications.

4 Conclusion

UNESCO states that there are around 6500 languages in the world. However, 90% of the languages do not appear on the Internet. In our globalized, modern world the tendency is to use the common languages of the developed world, such as English or French, for business, commerce, education, and any other information interchange.

The market for mobile applications is similar to the above scenario. Developers and companies attempt to meet their customers' expectations according to their demand. Customers of mobile applications are mainly distributed among developed countries, a factor which ultimately determines the languages of mobile applications. More than 99 percent of mobile apps are developed in English, Chinese, Spanish, Japanese, German, French, Korean, Portuguese, Italian, or Russian.

What about applications for the populations of less developed countries? How can they use mobile applications in their own languages?

The MilanApps project facilitates translations of the user interface text of software applications to local languages. We have created a portal for developers to work in collaboration with reliable translators. Developers of mobile applications will upload their user interface text to the MilanApps portal and reliable translators will translate them into their own language of expertise. Therefore, developers will have the capability of creating multilingual applications for mobile apps users.

These multilingual applications will, of course, mainly, be tailored for people living in less developed countries and written in their local languages. Since people understand their first language best, and are most comfortable speaking it, learning activities should mainly be carried out in this own language. The MilanApps project gives them this opportunity. People will be able to use software in their own language. Consequently, people will be able to cope with new technologies, at least mobile applications, much more easily.

Moreover, this project will not only help people who are living in developing or non-developed countries, but also help preserving minority languages. As minority languages are more widely spread in mobile applications, the awareness about them will also increase proportionally among communities.

The MilanApps project prototype is currently ready and working properly. The MilanApps portal will be published on TEDECO servers for developers and translators use. Our expectations for the future are that increasingly more reliable translators will join the portal to expand the language translation capacity. As the portal becomes more popular among developers and translators, our project will reach more users and achieve its goals.

4.1 What next?

In the future we expect that more translators will join the project. However, this situation may cause reliability problems related to the control mechanism of translators and translated text.

Translators will translate text into their own language of expertise. However, with an increased number of translators, it is possible that a 'fake' translator could ruin the translations with dummy text. In order to address this issue, we have created another role called "Reviewers". Reviewers will be responsible for checking the correctness of the translations. After their approval, we will then publish the translated text. This safety mechanism will keep the portal reliable by avoiding 'fake' translators.

A second problem could be caused by an excessive number of requests by translators. Currently, we have to check the CVs of translators to give them permission to translate text into their language. However, if a large number of requests come through the system, it would cause the permission mechanism to progress very slowly. Therefore, we are going to allow translators to login to the system through a reliable CV webpage, such as LinkedIn. This will help us check the translators' CV's and their reliability more easily.

References

Böhmer, M., Hecht, B., Schöning, J., Krüger, A., Bauer, G. (2011) 'Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage', in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, ACM: New York, NY, USA, 47–56, available: http://doi.acm.org/10.1145/2037373.2037383 [accessed 22 July 2014].

Cormack, M., Hourigan, N. (Eds.) (2007) *Minority Language Media: Concepts, Critiques and Case Studies*, Multilingual Matters Ltd: Clevedon ; Buffalo.

Github (2014) Build Software Better, Together [online], *GitHub*, available: https://github.com [accessed 22 July 2014].

IDE, A. (2013) PHP Just Grows & Grows | Netcraft [online], available: http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html [accessed 22 July 2014].

James, M. (2007) The SourceForge Story - Datamation [online], available: http://www.datamation.com/osrc/article.php/3705731/The-SourceForge-Story.htm [accessed 22 July 2014].

Kapetanakis, M. (2012) Developer Economics 2012 - The New App Economy - VisionMobile [online], available: http://www.visionmobile.com/blog/2012/06/report-developer-economics-2012-the-new-app-economy/ [accessed 22 July 2014].

Ludwig, S. (2012) Study: Mobile App Usage Grows 35%, TV & Web Not so Much [online], *VentureBeat*, available: http://venturebeat.com/2012/12/05/mobile-app-usage-tv-web-2012/ [accessed 22 July 2014]

Perez, S. (2012) 'comScore: In U.S. Mobile Market, Samsung, Android Top The Charts; Apps Overtake Web Browsing', *TechCrunch*, available: http://social.techcrunch.com/2012/07/02/comscore-in-u-s-mobile-market-samsung-android-top-the-charts-apps-overtake-web-browsing/ [accessed 22 uly 2014]

Portio Research (2013) Mobile Applications Futures 2013-2017 Market Report - Portio Research [online], available: http://www.portioresearch.com/en/mobile-industry-reports/mobile-industry-research-reports/mobile-applications-futures-2013-2017.aspx [accessed 22 July 2014]

Sheyholislami, J. (2009) 'Minority Language Media: Concepts, Critiques and Case Studies', *Canadian Journal of Communication*, 34(4), available: http://www.cjc-online.ca/index.php/journal/article/view/2287/ [accessed 22 July 2014].

Sqlite (2015) About SQLite [online], available: http://www.sqlite.org/about.html [accessed 22 July 2014].

Tethras (2013) Into Which Languages Should I Localize My Mobile App | Blog [online], available: http://blog.tethras.com/into-which-languages-should-i-localize-my-mobile-app/ [accessed 22 July 2014]

Unesco (2005) First Language First: Community-

Based Literacy Programmes for Minority Language Contexts in Asia | MTB-MLE Network [online], available: http://mlenetwork.org/content/first-language-first-community-based-literacy-programmes-minority-language-contexts-asia [accessed 22 July 2014]

W3C (2011) HTML5 Differences from HTML4 [online], available: http://www.w3.org/TR/html5-diff/ [accessed 22 July 2014]

Appendices

A. Technology Decisions

1. HTML
HTML5 is a core technology markup language used for structuring and presenting content for the World Wide Web. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML 4 as of 1997) (W3C 2011).
2. PHP
PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January, 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers (IDE 2013).
3. BootStrap
Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. In March 2014 it was the No.1 project on GitHub with over 65,000 stars and 23,800 forks (Github 2014) .
4. SQLite3
SQLite is an in-process library that implements a self-contained, server-less, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is currently found in more applications than we can count, including several high-profile projects (Sqlite 2015).
5. Android Sdk
The Android SDK provides API libraries and developer tools necessary to build, test, and debug apps for Android.

B. Use Cases

Reliable Translators: Reliable translators contribute to the MilanApps languagedatabase through the webpage.
MilanApps System: The MilanApps webpage.
Developers: Developers of mobile applications.

- They introduce UI text of their applications to the MilanApps System.
- They fetch the translated text after translators translate them.

Market: Server to download applications.
Users: Users of mobile applications. They can use the applications in their own language. (See Figure 3)

1. MilanApps Use Case Diagram

See Figure 4

Use Case UC1: Register with the System

Scope: MilanApps System

Level: Sub-function

Stakeholders & Interest:

-Translator: Wants to register with the system without any problem.

Preconditions: The translator is not registered.

Success Guarantee: The translator's record is saved

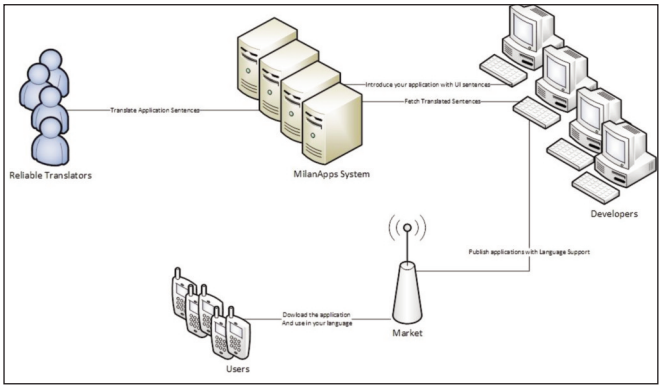

Figure 3. Basic Workflow Scheme of MilanApps

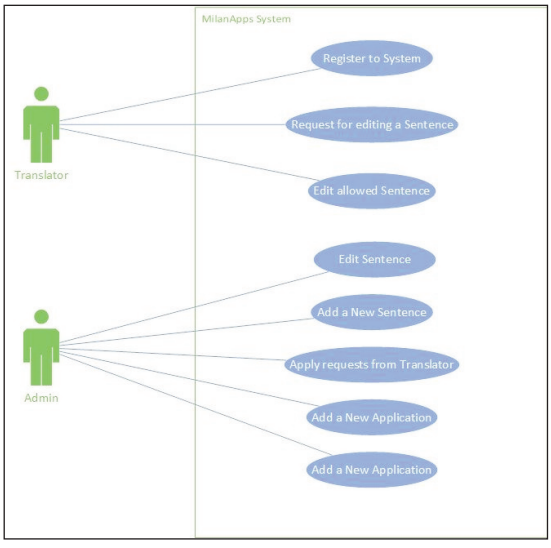

Figure 4: Use case diagram of MilanApps

and the translator is ready to login to the system.

Main Success Scenario:

1 The translator enters his/her e-mail address.
2 The translator enters his/her password.
3 The translator re-enters his/her password.
4 The translator submits the form.
5 The system records the form and sends an e-mail with an activation link to the translator's e-mail address.
6 The translator clicks on the link from his/her e-mail account.
7 The system updates the information for the translator.

Use Case UC2: Request for translating text

Scope: MilanApps System

Level: Sub-function

Stakeholders & Interest:

-Translator: Wants to request a language to translate text.

-Admin: Wants to receive requests from translators.

Preconditions: The translator's record is saved and the translator is ready to login to the system.

Success Guarantee: The translator's request is saved and sent to the admin.

Main Success Scenario:

1 The translator logs in to the system.
2 The translator enters a request for translating a language text.
3 The translator selects a language to translate.
4 The translator submits the form.
5 The system records the form and sends the information to the admin.
6 The system informs the user that the request is done.

Use Case UC3: Translate Text

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

-Translator: Wants to translate a language text.

Preconditions: The translator's record is saved and the translator is ready to login to the system.

Success Guarantee: The translator's changes are saved to the database.

Main Success Scenario:

1 The translator logs in to the system.
2 The translator enters the "translate a language" menu.
3 The translator selects the application.
4 The translator selects the reference language.
5 The translator selects the target language.
6 The translator submits the selections.
7 The system records the form and presents the table for translating.

8 The translator translates text.
9 The translator saves the changes.
10 The system updates the database according to the changes.

Use Case UC6: Edit translated text

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

-Admin: Wants to edit translated text.

Preconditions: None.

Success Guarantee: The admin's changes are saved to the database.

Main Success Scenario:

1 The admin logs in to the system.
2 The admin enters the "translate a language" menu.
3 The admin selects the application.
4 The admin selects the reference language.
5 The admin selects the target language.
6 The admin submits the selections.
7 The system records the form and presents the table for translating.
8 The admin edits translated text.
9 The admin saves the changes.
10 The system updates the database according to the changes.

Use Case UC7: Add a new language

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

- Admin: Wants to add a new language to the system.

Preconditions: none

Success Guarantee: The new language is saved to database.

Main Success Scenario:

1 The admin logs in to the system.
2 The admin enters the "add a new language" menu.
3 The admin enters the name of a new language.
4 The admin submits the form.
5 The system records the form and updates the database.
6 The system informs the admin that the process is done.

Use Case UC8: Apply Requests from Translators

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

- Admin: Wants to see the requests from translators and apply them.

Preconditions: The translators have sent a request.

Success Guarantee: The admin's response is saved and the translator is ready to translate the requested language.

Main Success Scenario:

1 The admin logs in to the system.
2 The admin enters the "requests from users" menu.
3 The admin reviews the requests.
4 The admin applies a request.
5 The system records the form and updates the database.

Use Case UC9: Add a new Application

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

- Admin: Wants to add a new application to the system.

Preconditions: None.

Success Guarantee: New application is saved to the database.

Main Success Scenario:

1 The admin logs in to the system.
2 The admin enters the "add a new application" menu.
3 The admin enters the name of a new application.
4 The admin submits the form.
5 The system records the form and updates the database.
6 The system informs the admin that the process is done.

Use Case UC10: Add a new text for translation

Scope: MilanApps System

Level: User goal

Stakeholders & Interest:

- Admin: Wants to add a new text to the system.

Preconditions: None.

Success Guarantee: A new text is saved to the database.

Main Success Scenario:

1 The admin logs in to the system.
2 The admin enters the "add a new text" menu.
3 The admin enters new text.
4 The admin submits the form.
5 The system records the form and updates the database.
6 The system informs the admin that the process is done.

C. MilanApps Database Structure

1. Table: Language

Entity 1: _id

- Description: Id number of the Language, each language has a unique Id number.
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

Entity 2: Name

- Description: Name of the Language.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

2. Table: Apps;

Entity 1: _id

- Description: Id number of the Application, each application has a unique Id number.
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

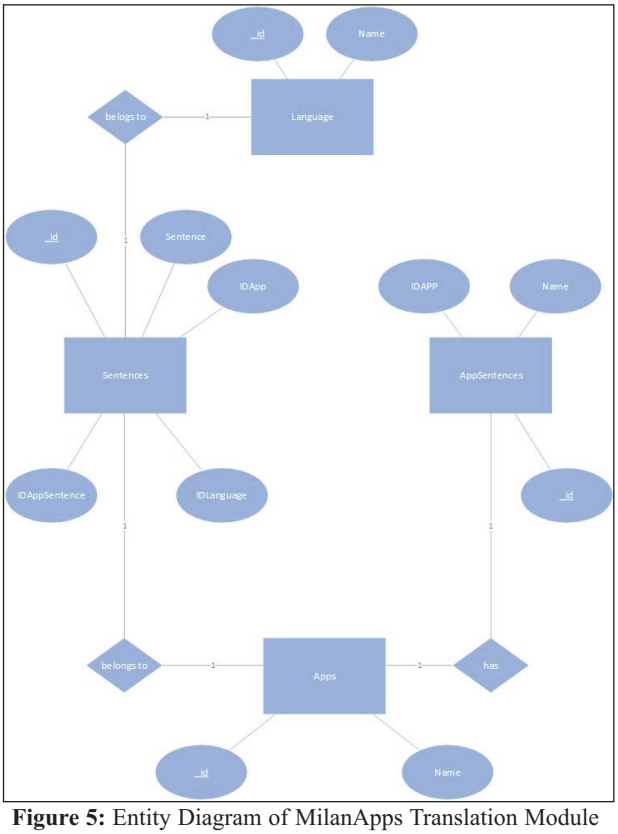

Figure 5: Entity Diagram of MilanApps Translation Module

Entity 2: Name

- Description: Name of the Application.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

3. Table: Sentences;

Entity 1: _id

- Description: Id number of the Sentence, each sentence has a unique Id number.
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

Entity 2: Text

- Description: Name of the Text.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 3: IDApp

- Description: Id Number of the Application.
- Data Type: INTEGER
- Primary Key: NO
- Foreign Key: YES
- Auto Increment: NO

Entity 4: IDLanguage

- Description: Id number of the language.
- Data Type: INTEGER
- Primary Key: NO
- Foreign Key: YES
- Auto Increment: NO

Entity 5: IDAppSentence

- Description: ID number of IDAppSentence.
- Data Type: INTEGER
- Primary Key: NO
- Foreign Key: YES
- Auto Increment: NO

.

4. Table AppSentences

Entity 1: _id

- Description: Id number of the AppSentences/
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

Entity 2: Name

- Description: Name of the IDAppSentence.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 3: IDApp

- Description: Id Number of the Application.
- Data Type: INTEGER
- Primary Key: NO
- Foreign Key: YES
- Auto Increment: NO

5. Database of MilanApps Translators

See Figure 6.

6. Table: Users;

Entity 1: ID

- Description: Id number of the User, each User has a unique Id number.
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

Entity 2: Email

- Description: E-mail address of the User, each User has a unique e-mail.
- Data Type: TEXT
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: NO

Entity 3: Password

- Description: Password of the User.
- Data Type: TEXT
- Primary Key: NO

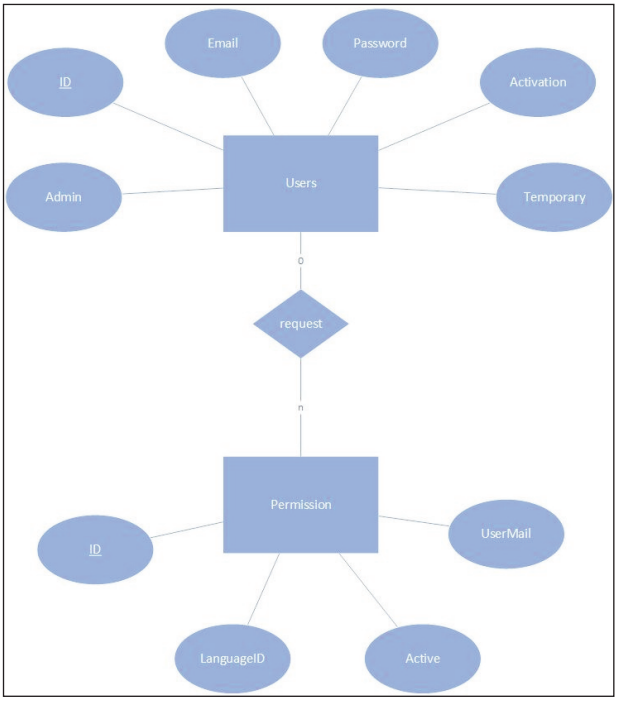

Figure 6: Entity Diagram of MilanApps Translators

- Foreign Key: NO
- Auto Increment: NO

Entity 4: Admin

- Description: Shows if the User is an admin or not.
- Data Type: BOOLEAN
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 5: Activation

- Description: Activation number which is sent to an e-mail address.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 6: Temporary

- Description: Shows if the user's account is activated or not.
- Data Type: BOOLEAN
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

7. Table: Permission

Entity 1: ID

- Description: Id number of the Permission, each permission has a unique Id number.
- Data Type: INTEGER
- Primary Key: YES
- Foreign Key: NO
- Auto Increment: YES

Entity 2: LanguageID

- Description: Language id of the request, each request has a language ID.
- Data Type: INTEGER
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 3: Active

- Description: Shows if the permission is given or not.

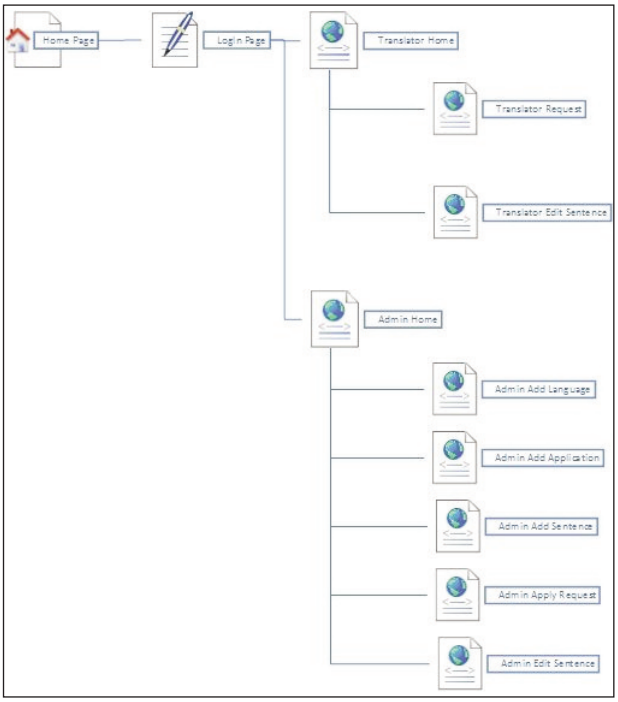

Figure 7: Website Map Diagram of MilanApps

- Data Type: BOOLEAN
- Primary Key: NO
- Foreign Key: NO
- Auto Increment: NO

Entity 4: UserMail

- Description: E-mail of the user who makes the request.
- Data Type: TEXT
- Primary Key: NO
- Foreign Key: YES
- Auto Increment: NO

8. MilanApps Website Map Diagram

See Figure 7

D. MilanApps Forms

1. MilanApps Translation Module for Translators

Login Page

This is the main page of the MilanApps website. If the translator does not have an account, he/she has to register with the system first.

If the translator has an account on the MilanApps system, he/she will enter his/her e-mail address and password to login.

- The translators will enter their username and password to enter the MilanApps System.
- They can register with the system by entering their e-mail address and password.
- The system will send them an activation e-mail to verify they are human.

Request Page

After the translator logs in to the system, he/she can translate and record text to his/her language of expertise. However, he/she needs to obtain permission for that language first. The translator can request a language for translation through this page. The translator's request will be accepted by the admin if his/her reliability is verified.

- The translators will send a request for a language to translate.
- The admin will verify the translator and if he/she is reliable, the admin will apply the request.

Editing Page

In this page, translators can translate and record text to his/her language of expertise. They need to obtain permission for that language from the admin in order to translate text.

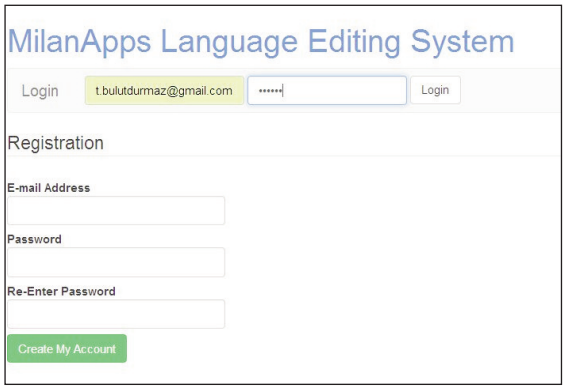

Figure 8: MilanApps Login Page

Figure 9: MilanApps Request Page

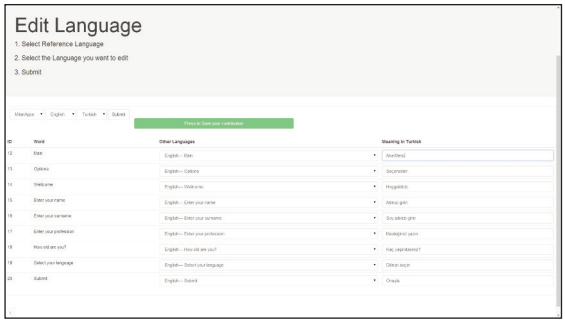

Figure 10: MilanApps Edit Language Page

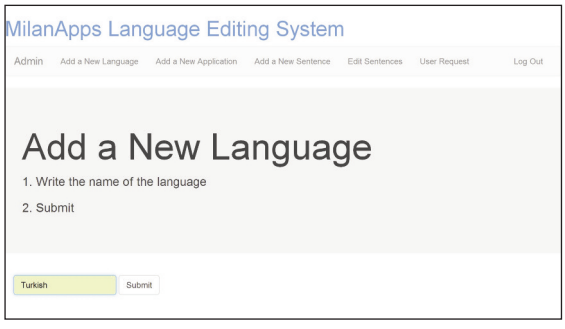

Figure 11: MilanApps Add a new Language Page

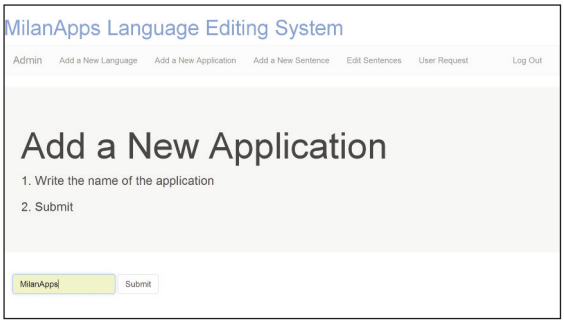

Figure 12: MilanApps Add a New Application Page

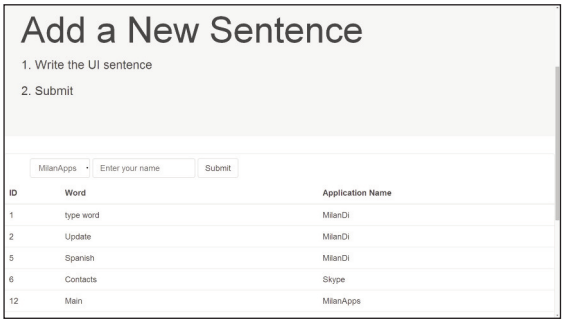

Figure 13: MilanApps Add new Text Page

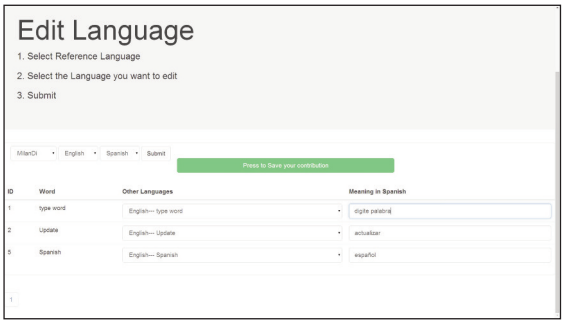

Figure 14: MilanApps Edit Language Page

Figure 15: MilanApps User Request Page

- Translators will select the application they want to contribute to.
- Translators will select the language they want to translate from.
- Translators will select the language to translate to according to their expertise.

2. MilanApps Translation Module for Admins

Add a new language Page
The admin can add a new language to the system from this page.

- The admin introduces a new language into the system.

Add a new application Page
The admin can add a new application to the system from this page.

- The admin introduces a new application into the system.

Add new text Page
The admin can add new text to the system for a specific application from this page. The UI text of the applications will be entered by the admin for the translators to translate them to their local languages.

- The admin selects the application to add text to.
- The admin enters the text.

Edit Language Page
In this page, the admin can translate and record text to any language. They are allowed to make any change.

- The admin will select the application they want to contribute to.
- The admin will select the language they want to translate from.
- The admin will select the language to translate to according to their expertise.

Figure 14: MilanApps Edit Language Page

User Request Page
The admin can apply the request from translators. If the admin gives permission to a translator's request, that person can translate text in that specific language.

- The admin applies requests from translators.

E. MilanApps Test Application

Figures 16, 17 and 18 show screenshots that belong to the test mobile application. The UI text of this application was entered into the MilanApps system. Translators translated the text to their own local languages. This application then fetched this new data to support more languages. Currently, this sample mobile application supports English, Turkish, Kurdish and Spanish.

Main Menu (shown in figures 16)
- This is the main menu of the application. Text is in English.

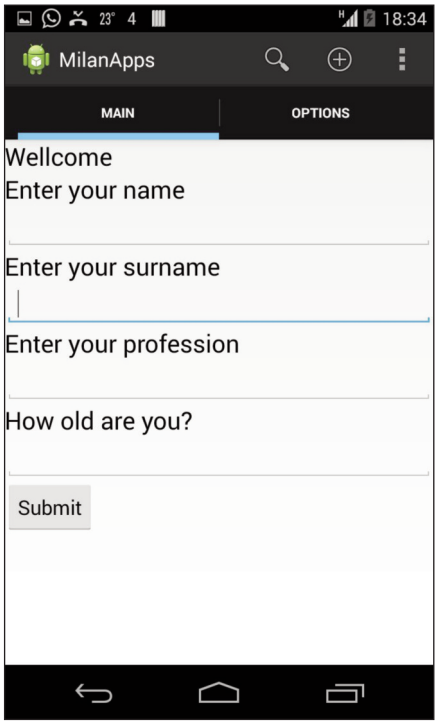

Figure 16: Main Menu of MilanApps Test Application

Options Menu (shown in figures 17 & 18)
In the options menu, users can select the language they prefer.

- User selects his language and submits
- The language of the application has been changed and the sentences are in Turkish now.

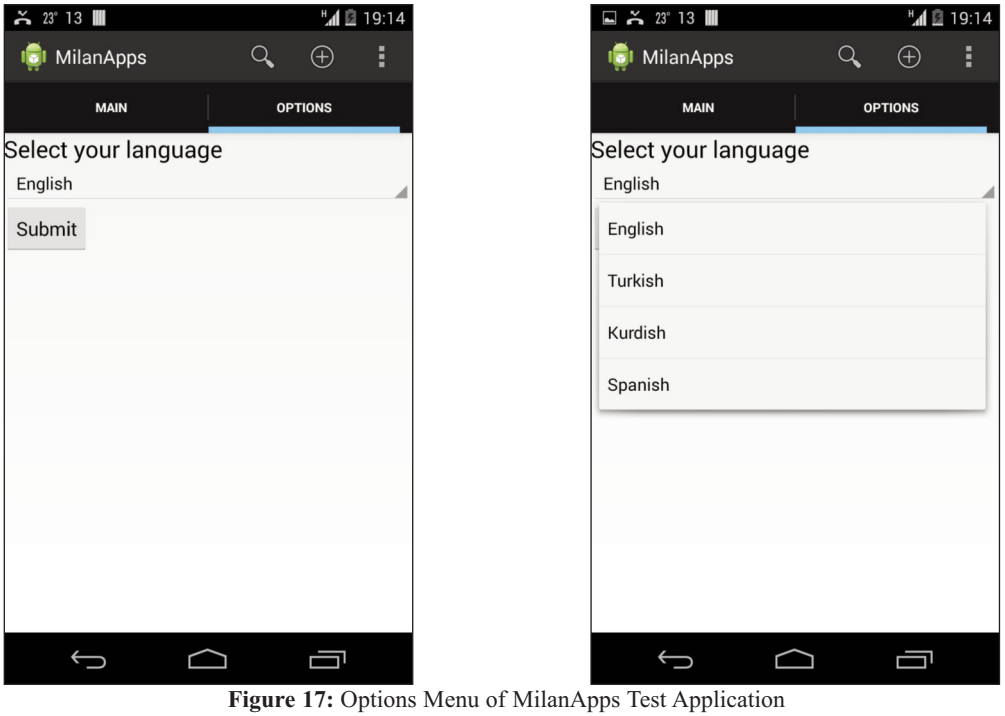

Figure 17: Options Menu of MilanApps Test Application

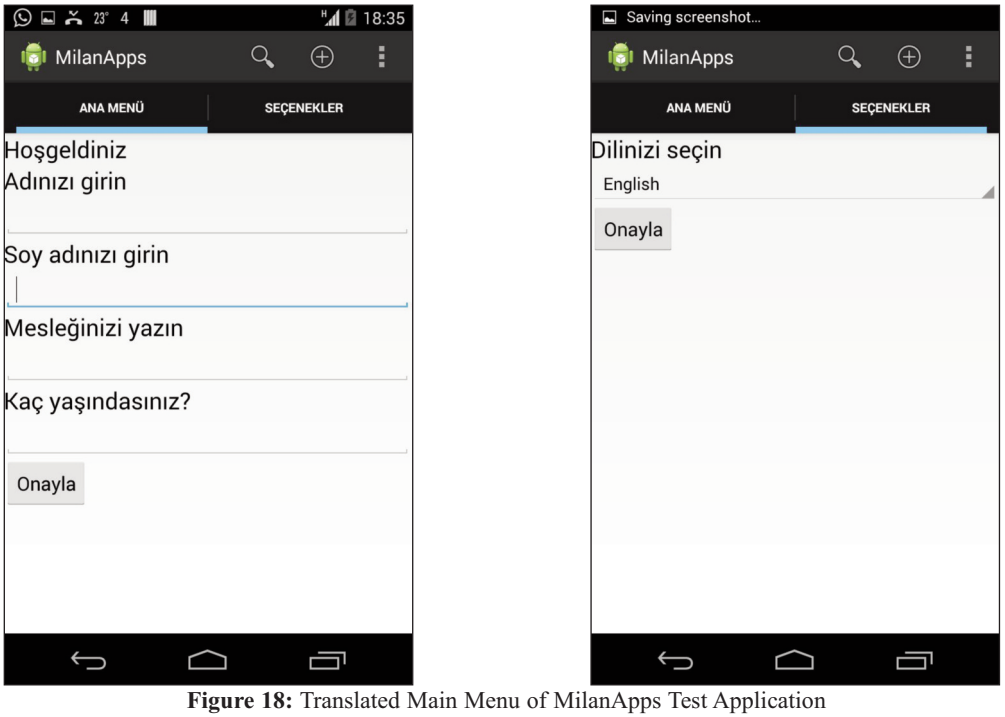

Figure 18: Translated Main Menu of MilanApps Test Application

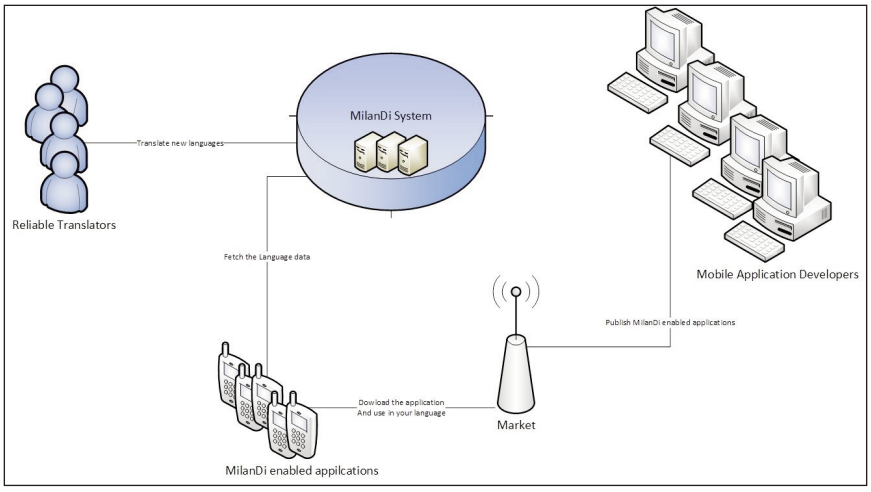

Figure 19: Workflow Scheme of MilanDi Project

F. MilanDi Project

1. Network Scheme

Reliable Translators:
Reliable translators are contributing Language MilanDi database through webpage (MilanDi Language Editing System).

MilanDi System:
MilanDi webpage which is Milandi Language Editing System

Mobile Application Developers:
Developers of mobile applications. They should build applications with MilanDi support.

Market:
Server to download applications

Milandi Enabled Applications:
MilanDi enabled applications can fetch the data from MilanDi system and able to convert languge datas according to user's requests.

2. MilanDi mobile dictionary

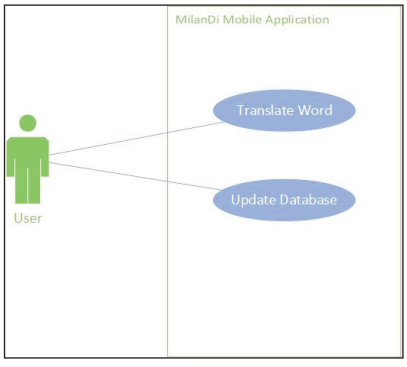

Figure 20: Use case diagram of MilanDi mobile dictionary

Use Case UC1: Translate Word
Scope: MilanDi mobile dictionary
Level: user goal
Stakeholders & Interest:
-User: Wants to translate the word
Preconditions: none
Success Guarantee: User reach the translated word
Main Success Scenario:
1 User picks the language he/she wants to translate from
2 User picks the language he/she wants to translate
3 User enters the word he/she wants to translate
4 System records the form and directs the result to the user

Use Case UC2: Update Database
Scope: MilanDi mobile dictionary
Level: sub function
Stakeholders & Interest:
-User: Wants to update the database of the application
Preconditions: mobile device is connected to Internet
Success Guarantee: Recent database is downloaded and embedded into application
Main Success Scenario:
1 User clicks update button on main page
2 System receives data from MilanDi server and embeds into application
3 System informs user

3 MilanDi Language Editing System

Use Case UC3: Register to System
Scope: MilanDi Language Editing System
Level: sub function

Stakeholders & Interest:
-Translator: Wants to register to system without any problem
Preconditions: Translator is not registered
Success Guarantee: Translator's record is saved and translator is ready to login to system
Main Success Scenario:
1 Translator enters his/her e-mail address
2 Translator enters his/her password
3 Translator re-enters his/her password
4 Translator Submits his form
5 System records the form and sends a mail with activation link to translators e-mail address
6 Translator clicks the link from his/her e-mail account
7 System updates translator information

Use Case UC4: Request for editing a Language
Scope: MilanDi Language Editing System
Level: sub function
Stakeholders & Interest:
-Translator: Wants to request a language to edit
-Admin: Wants to receive requests from translators
Preconditions: Translator's record is saved and translator is ready to login to system
Success Guarantee: Translator's request is saved and sent to admin
Main Success Scenario:
1 Translator logs in to the system
2 Translator enters request for editing a language menu
3 Translator picks a language to edit
4 Translator Submits his form
5 System records the form and sends a information to admin
6 System informs user that request has been carried out

Use Case UC5: Edit allowed Language
Scope: MilanDi Language Editing System
Level: user goal
Stakeholders & Interest:
-Translator: Wants to edit languages
Preconditions: Translator's record is saved and translator is ready to login to system
Success Guarantee: Translator's changes are saved to database
Main Success Scenario:
1 Translator logs in to the system
2 Translator enters edit a language menu
3 Translator picks reference language
4 Translator picks target language
5 Translator submits
6 System records the form and presents the table for editing
7 Translator edits word and descriptions
8 Translator saves the changes
9 System updates the database according to changes

Use Case UC6: Edit Language
Scope: MilanDi Language Editing System
Level: user goal
Stakeholders & Interest:
-Admin: Wants to edit languages
Preconditions: none
Success Guarantee: Admin's changes are saved to database
Main Success Scenario:
1 Admin logs in to the system
2 Admin enters edit a language menu
3 Admin picks reference language
4 Admin picks target language
5 Admin submits
6 System records the form and presents the table for editing
7 Admin edits word and descriptions
8 Admin saves the changes
9 System updates the database according to changes

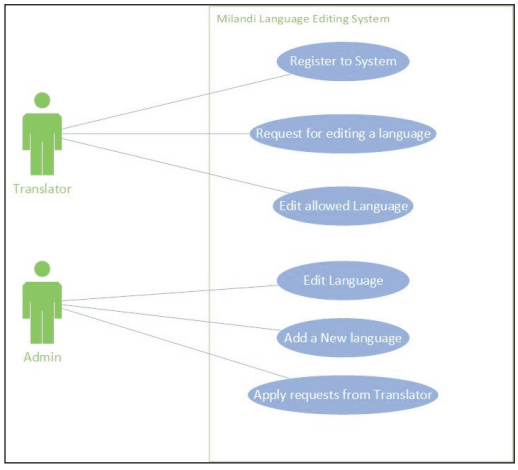

Figure 21: Use case diagram of MilanDi Language Editing System

Use Case UC7: Add a new language
Scope: MilanDi Language Editing System
Level: user goal
Stakeholders & Interest:
- Admin: Wants to add a new language to the system
Preconditions: none
Success Guarantee: New language is saved to database
Main Success Scenario:
1 Admin logs in to the system
2 Admin enters add a new language menu

3 Admin enters the name of a new language
4 Admin Submits his form
5 System records the form and updates the database
6 System informs admin that process is done

Use Case UC8: Apply Requests from Translator
Scope: MilanDi Language Editing System
Level: user goal
Stakeholders & Interest:
Admin: Wants to see the requests from translators and apply them
Preconditions: Translators had sent a request
Success Guarantee: Admin's response is saved and translator is ready to edit requested language
Main Success Scenario:
1 Admin logs in to the system
2 Admin enters requests from users menu
3 Admin reviews the requests
4 Admin applies the request
5 System records the form and updates the database

Guidelines for Authors

Localisation Focus
The International Journal of Localisation
Deadline for submissions for VOL 14 Issue 2 is 31 August 2015

Localisation Focus -The International Journal of Localisation provides a forum for localisation professionals and researchers to discuss and present their localisation-related work, covering all aspects of this multi-disciplinary field, including software engineering and HCI, tools and technology development, cultural aspects, translation studies, human language technologies (including machine and machine assisted translation), project management, workflow and process automation, education and training, and details of new developments in the localisation industry.

Proposed contributions are peer-reviewed thereby ensuring a high standard of published material.

If you wish to submit an article to Localisation Focus - The international Journal of Localisation, please adhere to these guidelines:

- Citations and references should conform to the University of Limerick guide to the **Harvard Referencing Style**
- Articles should have a meaningful title
- Articles should have an abstract. The abstract should be a minimum of 120 words and be autonomous and self-explanatory, not requiring reference to the paper itself
- Articles should include keywords listed after the abstract
- Articles should be written in U.K. English. If English is not your native language, it is advisable to have your text checked by a native English speaker before submitting it
- Articles should be submitted in .doc or .rtf format, .pdf format is not acceptable
- Excel copies of all tables should be submitted
- Article text requires minimal formatting as all content will be formatted later using DTP software
- Headings should be clearly indicated and numbered as follows: 1. Heading 1 text, 2. Heading 2 text etc.
- Subheadings should be numbered using the decimal system (no more than three levels) as follows:

  Heading
  1.1 Subheading (first level)
  1.1.1 Subheading (second level)
  1.1.1.1 Subheading (third level)
- Images/graphics should be submitted in separate files (at least **300dpi**) and not embedded in the text document
- All images/graphics (including tables) should be annotated with a fully descriptive caption
- Captions should be numbered in the sequence they are intended to appear in the article e.g. Figure 1, Figure 2, etc. or Table 1, Table 2, etc.
- Endnotes should be used rather than footnotes.

More detailed guidelines are available on request by emailing LRC@ul.ie or visiting www.localisation.ie

Localisation Focus
The International Journal of Localisation

Vol. 13 Issue 1 (2014)

CONTENTS