Localisation Focus

THE INTERNATIONAL JOURNAL OF LOCALISATION

ISSN 1649-2358

The peer-reviewed and indexed localisation journal

VOL. 9 Issue 1

EDITORIAL BOARD

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/resources/locfocus/pdf.htm and click on the issue you wish to download. Use the following logon details - username: locfocsub and password: V90102

Members of **The Institute of Localisation Professionals (TILP)** receive Localisation Focus – The International Journal of Localisation as part of their membership benefits. Membership applications can be filed electronically from ***www.tilponline.org*** Change of address details should be sent to LRC@ul.ie

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

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.

A CHALLENGE - FROM THE EDITOR

Tempora mutantur, loosely translated as "The Times They Are a-Changin'" by Bob Dylan in 1964, captures the essence of what was 2010. Change is written all over the annual reviews and highlights of 2010.The turn of fortune in the economy will be hard to take for many people and businesses. However, as with all dramatic changes it will also offer tremendous opportunities, and that includes localisation.

So, maybe it is about time to change not localisation itself, but the way it is dealt with by practitioners and perceived by the public. Maybe it is time to correct what is perhaps the most persistent myth in localisation, i.e. that localisation always requires a business case. This slightly dated concept from the 1980s has long been proven wrong by the world's most successful entrepreneurs. It has also been proven wrong by millions of people around the globe involved in "making the world a better place", people who localise for good, *for progress and for humanity, to make access to knowledge in every language a reality.*

In this issue, **Lenker, Anastasiou** and **Buckley** report on their research into workflow specifications and a representation framework for such specifications for enterprise localisation. If you agree with the authors, localisation has reached a level of maturity that will allow it to achieve what other fields, such as manufacturing and accounting, achieved back in the 1970s: an optimisation and standardisation of processes leading to continued process improvements. Their article provides solid arguments for a change in the perception that localisation processes cannot be standardised.

Localisation is now ubiquitous in the developed world. You come across it not just when surfing the web or when playing with your smart phone. You also come across it when you drive your car where a localised driver information system (DIS) will allow you to check the time, mileage (or kilometres), speed, the average consumption of fuel, next fuel stop due and so on. **Olaverri-Monreal, Bengler, Breisinger** and **Draxler** cover everything you ever wanted to know about how to internationalise and localise a DIS for drivers across the world.

Jiménez-Crespo will *change* your perception that internationalisation and localisation will always involve different languages. He discusses the strategies employed by multinational corporations to create an international version of their web content for a language that is spoken in different countries and cultural regions. He does not deal with English, perhaps the most obvious choice of language in this context, but with Spanish which to him presented a more interesting case because of its more varied use around the world.

One of the standard-lessons we have been trying to teach international developers is to avoid the use of concatenated strings at all cost. Many localisers have spent much of their professional life trying to convince English-speaking developers that a string such as "<SUBJECT>+<VERB>+<OBJECT>" to be instantiated as, for example, "You"+"have" +"2"+"messages" works well in English, but not necessarily in other languages. Yet, when localising a text-based online game into German, Spanish and Russian, **Arthur, Fedane, Brandt** and **Hannan** were faced with a system whose interface was based almost exclusively on string concatenation. Their report shows how one can localise software successfully, even if some of us would have sent it straight back to the developers telling them to sort out at least all the major internationalisation issues before submitting it again for localisation.

One of the constants in text book localisation strategy that is about to change is that localisation decisions are made by corporate headquarters. For a long time, corporate HQ decided which digital content should be made available to whom in which language, and when. **Exton, Spillane**, and **Buckley** report on the Babel Software project, a micro-crowdsourcing implementation that will allow users to take charge of that localisation decision. "Give up the illusion of control", advice given by Greg Oxton to those in charge of customer support is now being extended to those taking corporate localisation decisions.

Dylan never translated the full original hexameter *Tempora mutantur*, *nos et mutamur in illis,* meaning "The Times They Are a-Changin and so do we". Let's change in 2011 and demonstrate to society (not just to our shareholders) what we localisers can do to make the world a better place! Now here is a challenge.

Markup Languages and Menu Structure Transformation during the Internationalisation process of Driver Information Systems

Cristina Olaverri-Monreal[ab], Klaus-Josef Bengler[bc], Marc Breisinger[b], Christoph Draxler[d]

[a] Instituto de Telecomunicações, DCC/FC - University of Porto, 4169-007 Porto, Portugal

[b] BMW Forschung & Technik GmbH, Hanauerstr. 46, 80992 Munich, Germany

[c] Faculty of Mechanical Engineering, Technische Universität München, D-85748 Garching, Germany

[d] Institute of Phonetics and Speech Communication, Ludwig-Maximilians-Universität, 80799 Munich, Germany

cristina.olaverri@dcc.fc.up.pt

Abstract

Driver Information Systems (DISs) are used in a mobile environment, which implies that the ease of communication between driver and system is determined by the navigation between different options from the graphical user interface. The way a driver interacts with a menu can differ among cultures, which can lead to decreased fluency when navigating through an insufficiently localised menu structure. Therefore, menu structures of a DIS should be subjected to localisation, and software should be internationalised to allow a variable menu structure. This paper shows an implementation approach that facilitates the adaptation of menu structures in DISs. We use XML to represent a common menu structure which can subsequently, and independently, be transformed into new versions for each culture using XSLT Stylesheets. With this approach we can not only achieve separation of the content of the menu from its structure, but also the possibility of varying the structure itself without changing the basic program version.

Keywords: *internationalisation, markup languages, driver information systems, localisation, XML, XSLT, graphical user interfaces, human computer interaction, menu*

1. Introduction

One of the most important characteristics of current economic development is globalisation, i.e. the worldwide expansion of markets. *Globalisation* promotes *localisation,* which is the adaptation of products to different cultures. Localisation becomes more easily achieved if it has already been taken into consideration early in the development of a product, a process known as *internationalisation.* This leads to a product that can be adapted to a target market with relatively minor changes facilitating the localisation process significantly and reducing costs (Esselink 2000, Tuffley 2003).

Internationalisation is especially applicable to software. A generalised source version can be implemented in a way that allows subsequent changes to be automated. In contrast, the localisation of finished but non-internationalised software can require the implementation of new, different versions.

1.1 Driver Information Systems

Driver Information Systems (DISs are multifunctional systems that are used for entertainment and the display of vehicle information (Dewar and Olson, 2007, Rößger 2004). Traditionally they consist of several subsystems, such as audio and navigation systems or telephone functionality. However, because of the comparably low cost of the hardware, additional functionality is increasingly included in these devices and DISs have become very widely distributed. This makes the

adaptation of DISs to international markets especially important.

DISs are used in a mobile environment which implies several particularities for localisation. For example, the limitation of the monitor size requires a special layout and text to fit in the available space. During the development stages one must consider whether the translated text fits into the space provided (Musale 2001). Insufficient space requires additional localisation engineering work (Pérez González 2009). In addition, programming languages that are compatible with wireless technologies are frequently used for implementing DISs. For the localisation of systems that are designed for road use different legal rules have to be taken into account. For example, contrary to most states in the USA, in most European countries the use of mobile phones without headsets while driving is not permitted.

An additional point that has to be considered when localising DISs is the potential distraction these devices could cause when driving a car. It has been shown that poor interfaces can avert visual attention from the road (Tijerina et al. 2000), which has obvious negative effects on driving performance and might cause crashes; a finding that has been supported by several authors (Dewar and Olson 2007, Goodman et al. 1999, Lee et al. 2007, Rakauskas et al. 2004). Although no crashes have been directly connected to DISs, multiple glances between the device and the road have the potential to reduce drivers' ability to maintain vehicle control. In principle, the use of DISs can delay or interrupt cognitive processing of road-related information, resulting in longer reaction times (Alm and Nilsson 1994, 1995, Lee et al. 2001).
DISs that have a screen as a user interface can be a source of distraction because the driver has to move their eyes from the road in order to communicate with the DIS.
Thus, the localisation of DISs must be carried out in such a way that the localised version does not carry an increased risk of potential distraction for the driver.

First priority when trying to increase driving safety is to organise the menus in the display in such a way that they represent a minor distraction and facilitate the interaction between the user and the system. The localisation of menu structures can provide increased user friendliness for DISs menus (Heimgärtner 2007). For example, the use of satellite radio in the United States is widely distributed, thus, positioning this function prominently in the menu structure facilitates menu navigation to this frequently used function. However in Europe, where satellite radio is rarely used, to situate such a function in a prominent position would be rather distracting.

There are certainly functions within DISs that are less distracting than others. Several studies focused on radio use in traffic. Both, manual radio-tuning and manual dialling were found to influence the regulation of speed and thus can be disruptive to driving (Tijerina et al. 1995). In addition, crash data indicates radio-tuning is associated with crash involvement (Wierwille and Tijerina 1996). While the disruptions related to the physical actions of manual dialling and manual radio-tuning are small or nonexistent they can, nonetheless, influence driver behaviour (Green et al. 1993, Hayes et al. 1989, Tijerina et al. 1995).

The magnitude of demands on visual attention while dialling is sometimes less than that associated with manual radio-tuning (Hayes et al. 1989); while at other times, dialling may require a greater number of glances and increase the total time that the eyes are off the road (Tijerina et al. 1995). Cognitively demanding voice communications also appear to increase driver brake- reaction times, again indicating a reduction in situation awareness (Tijerina et al. 2000). However, a smooth interaction between human and machine reduces the number of glances away from the road and the development of a menu structure that can be adapted to specific culture preferences and habits improves this interaction. An additional interesting remark is that the value of cognitively demanding voice communications decreases under challenging traffic situations. (Draxler et al. 2001) showed that the effects that can be observed under higher workload produce an increased rate of misspellings and hesitation phenomena in the interaction between driver and machine.

Currently, most DISs are organised in the form of menu hierarchies. The fluency of communication between driver and system is determined by the kind of navigation between different options from the graphical user interface. Development of DISs without proper consideration of component integration can increase the cognitive load, errors and annoyance for drivers (Lee and Kantowitz 2005). Thus, fluency in navigation through the menu improves driver concentration on the road, in contrast to a poorly designed menu structure detracting from

driver concentration. Currently only terminology aspects are considered when localising a DIS; that the menu structure should also be adapted to the target culture is, in general, not taken into account.

1.2 Current internationalisation approaches

According to Taylor (1992) there are three possible methods of internationalisation:

- one can, in each case, write the code for the entire program in the language of the target culture;
- one can store all linguistic factors and cultural resources in separate files;
- one can allow the user to specify the target language. This involves integrating all possible variants in one system.

However, separate storage of linguistic and cultural resources is preferred in newer work. So in order to achieve the internationalisation of a software-based product such as the DIS it is common to have the aim of separating the storage of basic elements, which are fixed for all countries, and of variable elements, such as texts, etc. (Hudson and Hall 1997, Rößger 2003). In the localisation phase these elements are then united. According to (Rößger 2004) the software should be developed in an internationalised way and then only in a second phase should the localisation for the individual markets take place. The vehicle contains, in principle, the same hardware, while the country-specific aspects are realised via software (Rößger 2003).

2. Implementation Approach

In this paper we present an implementation approach that facilitates the adaptation of menu structures in DISs.

For the correct use of this approach a previous international study of user preferences, regarding the location of menu options in a DIS, should be performed. This study will determine which options should be included in the different local versions.

We will describe an example of a menu structure for a DIS that is independent from the individual content. This separation is achieved by the use of Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT) and Scalable Vector Graphics (SVG). Thus it becomes possible to use several languages and local standards for a language or a country (character set, date and time format) can be built into each XSLT, which generates the representation for the respective country. The arrangement of the elements can be changed in XSLT using the "sort" function. Also font size and field length can remain variable and be programmed by a function, which is particularly useful and necessary for localisation into other writing systems (Esselink 2000). Separate source files for all elements of the user interface were built, so that the text of the individual buttons and the layout could be stored in separate files. With this method only minimal changes to the code are necessary in order to accomplish a fast localisation of the menu.

We use XML to represent a common menu structure based on a menu tree, which can subsequently be transformed into new target menu trees for different cultures using XSLT Stylesheets. With this approach we achieved not only separation of the content of the menu from its structure, but also we gained the ability to vary the structure itself without changing the basic program version.

2.1. Background information on the programming languages used

Markup languages such as HTML (Hypertext Markup Language), SGML (Standardised General Markup Language) or XML (Extensible Markup Language) are very useful in representing menu hierarchies because they include marking signs for data description (Eckstein and Casabianca 2001, Sperberg-McQueen et al. 2000, Whitehead et al. 2002). These marking signs can be used to define options and sub options hierarchically.

2.1.1 XML

XML was developed by the World Wide Web Consortium (W3C). In principle XML is used to describe structures but also contains notations for the specification of formal grammars in addition to data marking (Abiteboul et al. 1998, Neven and Van den Bussche 2002). As a meta language XML explains or defines other languages. Its elements describe only the text structure and not the output, contrary to HTML tags (Nussbaumer and Mistlbacher 2002).

In an XML document the text parts of the document that are annotated using tags (called elements in XML) are ordered in a hierarchical form which makes it possible to represent each element's information in a graphical as well as in a formal way, in tree form (Lobin 2000). Therefore, XML permits the creation of structured documents. In addition, using a transformation language like XSLT, new XML documents can be created from XML documents and document parts.

2.1.2 XSLT

XSL (Extensible Stylesheet Language) is the language for formatting XML documents for the output. It allows the separation of content and format (van Otegem 2009); it is based on XML and represents the XML equivalent to Cascading Style-Sheets (CSS) for HTML (Savourel 2002). XSLT (Extensible Stylesheet Language Transformations) (Adler et al. 2000) is one of the components of XSL necessary for this transformation (Yu et al. 2005). It is tree-oriented and uses Template Rules for defining the rules for the transformation process. This allows the manipulation of an XML document in a similar way to a digital library (Gueye et al. 2004). Therefore, XML can be considered as extraordinarily appropriate as a multilingual format for data (Bradin 2002), and also especially appropriate for the definition of variable structures.

2.2. Implementation

2.2.1. First Transformation

A basic menu structure, which formed the base for the subsequent transformations, was designed in XML. All possible states of the options that had to be varied were implemented in the basic menu structure. The menus for Germany, the United States and Japan represented the basic menu structure. This structure was modified through transformations at eight nodes into the three respective versions.

The XML documents were created according to the XML Schema definition in Figure 1 and the elements "main menu", "menu" and "menu item" were defined. They were used to describe the tree structure in all of the XML documents that were created. The element "main menu" represented the point of origin of the tree structure and constituted the "Root" element of the XML document while all elements of lower hierarchies were designated with the name "menu" or "menu item". The element "menu item" described a terminal tree node in a hierarchy level, and options where a function would be executed. Figure 2 shows an example of this structure as XML code.

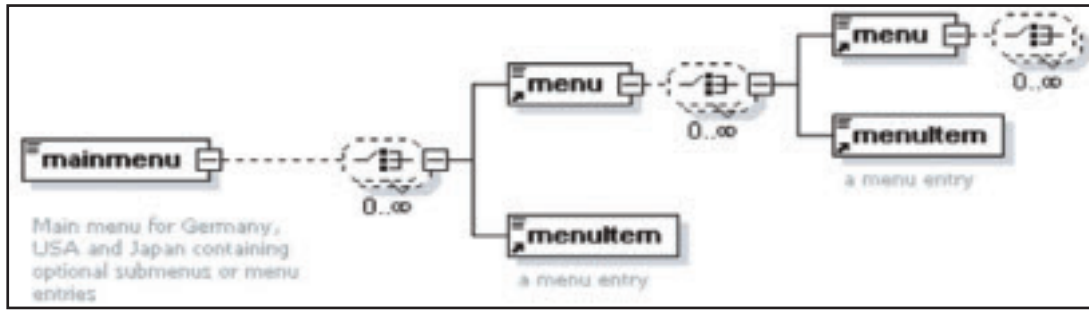


Figure 1. Schematic definition for describing the tree structure of the XML documents

We used three transformations to produce the localised menu structures from an internationalised version. During the first transformation the different positions of the options within the menu were defined. The text elements were included during the second transformation, and the third transformation was used to create SVG files to be visualised.

```
<menu sid="SID_ENTERTAINMENT">
    <menu sid="SID_FM">
        <menu sid="SID_FMSTOREDSTATIONS">
            <menuItem sid="SID_FMSTATIONSLIST"/>
        </menu>
        <menu sid="SID_FMRECEIVABLESTATIONS">
            <menuItem sid="SID_FMSTATIONSLIST"/>
        </menu>
    </menu>
    <menu sid="SID_AM">
        <menuItem sid="SID_TRAFFICINFO"/>
</menu>
```

Figure 2. Example of XML code showing the menu option with the ID "SID_ENTERTAINMENT"

The transformation into structures for each country was implemented by accessing the tree nodes of the basic XML document using an XSLT Stylesheet. For the German version, the basic structure was transformed by placing the options "FM", "Music" and "Traffic informatio" into the menu "Entertainment". For the US version, the menu "Entertainment" contained: "FM", "AM", "Sat radio" and "Music" whereas in the Japanese version it contained "FM", "AM", "TRF radio", "Music", "Video" and "Traffic information". The decision as to which options were to be included in the localised versions was made after the results of an online questionnaire that was part of a PhD Thesis (Olaverri Monreal 2006).

The XSLT transformation was performed by selectively copying the content of the XML document into a new XML document which represented the localised structure. Pattern matching was used to identify the variable options for the copying process. Changes in structure were achieved by leaving the respective options out. Structure parts that remained unchanged were copied as an entire sub tree.

2.2.2. Second Transformation

The XML documents created by the XSLT transformation were used as source documents for a second transformation into documents containing the respective text for each language. Separate XML documents (dictionaries containing the text elements for each language) were created for this purpose and their respective texts were translated. The source XML documents, as well as the dictionaries, contained an ID attribute for defining the node, or the text element that had to be inserted in the node respectively.

The XSLT document was used to compare the ID attributes of both documents and insert the corresponding translation into the target XML document. For this purpose variables containing the whole dictionary and the ID attribute of the matched element were used. In this way the program can load a locale-specific resource from the appropriate resource file for the current user's locale, in this case the XML files containing the translated texts. The results of the second transformation were, again, XML files containing the respective localised versions. These versions had been used for visualisation.

2.2.3. Third transformation

Visualisation of the menu structures of the localised versions was performed in the course of this study as a simulation of a DIS. The graphical user interface was implemented using the programming language SVG.

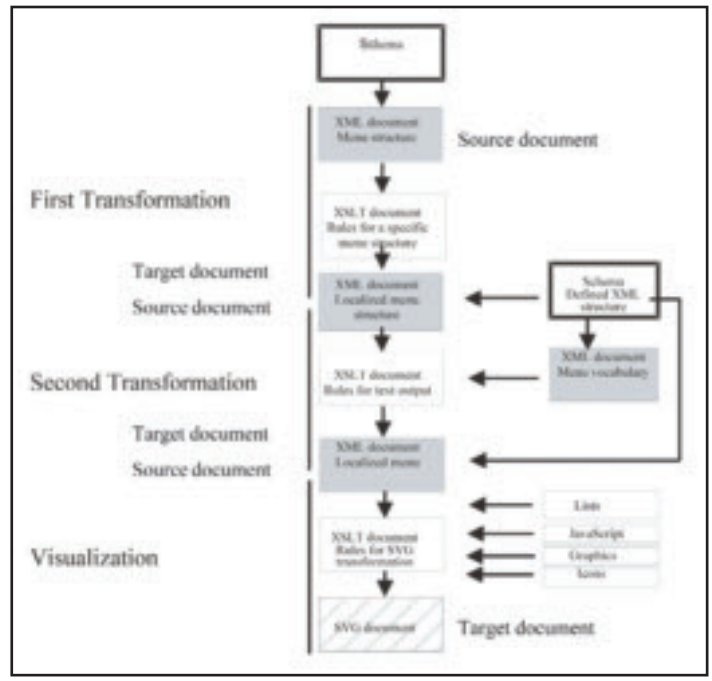


Figure 3. Internationalisation process of the DIS menu structure

The localised XML documents were transformed by additional XSLT documents into SVG documents **(Figure 3)**. SVG is a vector graphic format that is based on XML and is a W3C recommendation. The vector information is not saved in binary form but rather as text, in the form of XML code. Therefore, it can be changed, read and written without requiring that special programs be used (Spona 2001). As a result SVG is a very appropriate format for the graphical representation of XML structures.

This transformation was performed to visualise the tree structures in the form of a menu dialogue. Every single menu element of the SVG document that was to be visualised was stored in separate SVG files and defined through the corresponding XSLT documents. Single SVG files represented the different menu screens for the single country version.
Lists, for example radio stations, were embedded in the transformed XML document extracting them from their respective files. To define the position of the single elements and include functionality, JavaScript was used.

In the upper line of all SVG documents the keyboard input and rules for focusing the different menu options were specified. After this the script was loaded. The following SVG types were created:

- SVG files that began with the ID of the respective nodes (prefix "SID"), for example "SID_CLIMATE.svg". These files contained the submenus for the respective menu option (in this case climate). The standard icons with those for the submenu options were loaded first and consequently the background was generated (Figure 4). The background consisted of a large rectangle, a menu bar at the bottom of the screen, the icons "country" & "menu", the buttons for "play" & "stop", and the symbols for scrolling. After this background, the upper menu bar with the name of the main menu ("Top Menu"), the groups for focusing the options and the buttons with the sub-options of the corresponding menus were generated.
- Lists that were represented by SVG files with the prefix "FOL" (for example lists of songs) were also created. In the documents that had the following string "FOL_" as a part of the name, the icons were loaded and later the background and the upper menu bar. Then, the chosen option was brought into focus and the song title was shown. Finally, the focusing and defocusing subroutine of both buttons "Play" and "Stop" was called.

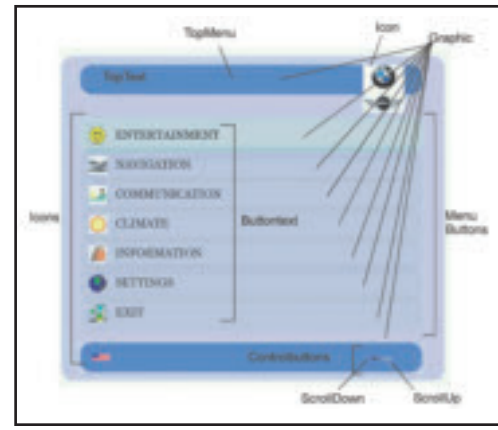


Figure 4. Elements of the Graphical User Interface that were defined in the JavaScript. The names of the elements correspond to the names in the XSLT templates

The XSLT document that was created for the visualisation contained seven templates representing the buttons and the menus for accessing the nodes and the text of the XML document. (For example an SVG document was created for the main menu options and buttons were created for the different options of this menu (Entertainment, Navigation, Climate and Exit)). Other XSLT templates created the buttons for controlling the music or radio (Play, Stop), and accessed the XML documents containing lists of songs or radio stations. In the latter case, a button was created and the element names of the list were shown. Menu options containing a submenu were visualised in an additional template creating just buttons for menu navigation. A different template contained the rules for the universal layout of the user interface. In addition global variables and JavaScript for specific layout according to localisation were used.

3. Discussion

In this paper we present a procedure to implement an internationalised menu structure for DISs using Markup languages. This structure can be adapted to different local versions by external files which allow the maintenance of the source code.

As previously mentioned, the development of internationalised software, i.e. of software that includes the main features relevant for the future localisation process, follows several requirements and can be summarised as the separation of source code and elements that are to be varied during the

localisation process (Esselink 2000).

Separation of variable elements and code is traditionally achieved by storing layout elements from the software code in different files and writing program code that is independent of the user's locale storing the locale-specific information in resource bundles.

We use XML for representing a common menu structure, which can subsequently and independently be transformed into new versions for each locale using XSLT Stylesheets. With this approach we have achieved not only the separation of the content of the menu from its structure, but also the possibility of varying the structure itself without changing the basic program version.

The main advantage of this approach is that the basic structure is thereby, in principle, extendible, so that modification can be taken into account. In this way a program can be created without requiring the full information about the element position and later modifications can be performed easily. For instance, as a usability test, a unique XML structure can be used that can be modified through several XSLT documents. In addition, new functions are continuously being developed for existing DISs that have to be included in the system. Therefore, it is feasible to implement a basic expandable structure which is transformed by XSLT and allows for the easy inclusion and adaptation of new elements, in this case, menu options.

One of the difficulties in the localisation process is the use of different platforms, data formats and character sets of non English software product versions. Sometimes an application that was designed in English can only be localised on another platform, version or environment in an Asian language. XML is platform independent and thus makes it possible to save documents and data in a structured form. This standard allows for a smooth document exchange. Furthermore, as Unicode is the default character encoding in XML complete multilingual compatibility is guaranteed (Bradin 2002). Because of these characteristics, XML is ideal for the creation of international documents.

The programming languages used for the implementation of the DIS (XML, XSLT, SVG and JavaScript) feature several characteristics that differentiate them from other, traditional, programming languages. With traditional programming languages it is difficult to achieve a varying menu structure within a source document.

Markup languages such as XML enable differentiation between the different structure levels since it includes annotations for data description (Eckstein and Casabianca 2001, Sperberg-McQueen et al. 2000, Whitehead et al. 2002). These annotations can be used to define options and sub-options in a hierarchical way. Using the defined templates with the template rules of the transformation language XSLT new XML documents can be created from existing XML documents and document parts. This technique was applied for transforming the XML source document into a target document with a different tree structure. Through a further XSLT document the text for each language was inserted.

Visualisation of the menu structures of localised versions was performed using the programming language SVG.
SVG is, like XSLT, a XML based language that is appropriate for representing graphical elements. It is also platform and application independent. The use of SVG in combination with XML and XSLT is extended. For example, (Arun and Ganguly 1999) proposed the use of SVG and VRML (further XML based languages) for the data description and its further transformation. (Arun and Ganguly 1999, Baravalle et al. 2003) created data in XML and transformed it with XSLT to create a SVG representation of scientific data. With this technique independence between data and representation (elements' position, colour, visual representation, etc.) as well as interoperability is reached. In addition, the probability that different devices or software applications can communicate directly among them increases. SVG can also be combined with JavaScript, in this way adding to a document that can be visualised for further functionality. Because of the platform independence of JavaScript, like the other used languages, complete independence of the internationalised system was guaranteed.

An important factor to consider when driving is distraction resulting from technology that draws the eyes and takes the driver's attention away from the road (Alm and Nilsson 1994, 1995, Lee et al. 2001). The main advantage of our approach is that the menus of the local version are organised in such a way that they take into account the cultural background of the driver. This structure facilitates the interaction between the user and the system, thus

increasing fluency and reducing the number of glances between the system and the road.

4. Conclusion

The collected data built the base for the implementation of a DIS simulation as an example of the internationalisation process focusing on the menu structure. A programming technique was proposed for facilitating the subsequent localisation of the menu structure that would be less time-consuming.

Adhering to internationalisation requirements implies using a special solution through variants. The solution based on the combination of XML, XSLT and SVG turned out to be the most appropriate because with XML it is easy to develop a variable Graphical User Interface with relatively little code (Rößger 2003).

In the case of requirements that can only be fulfilled using higher programming languages, a larger effort is required for representing variants (i.e. in the dialogue, menu or in the graphics). This can be achieved by designing a complete system for each variant or programming the necessary extensions. The solution suggested in this paper considers variants already present in the basic structure. It can be combined with other technologies and programs, so that the service programs that are necessary for the DIS functions can be integrated.

As a result of the extended use of XML during the DIS development (Rößger 2004) and the hierarchical menu structure, the creation of XML variants through XSLT constitutes a useful method for achieving the internationalisation requirements of a menu structure.

References

Abiteboul, S., Cluet, S. and Milo, T. (1998) 'A logical view of structured files'. The VLDB journal, 7, 96-114.

Adler, S., Berglund, A., Caruso, J., Deach, S., Grosso, P., Gutentag, E., Milowski, A., Parnell, S., Richman, J. and Zilles, S. (2000) 'Extensible Stylesheet Language (XSL), Version 1.0'. W3C Working Draft [Online]. Available: http://www.w3.org/TR/2000/WD-xsl-20001018/.

Alm, H. and Nilsson, L. (1994) 'Changes in driver behaviour as a function of handsfree mobile phones-a simulator study'. Accident Analysis and Prevention, 26, 441-452.

Alm, H. and Nilsson, L. (1995) 'The effects of a mobile telephone task on driver behaviour in a car following situation'. Accident Analysis & Prevention, 27, 707-715.

Arun, B. and Ganguly, A. D. (1999) 'XML to be, VRML to see'. Crossroads [Online], 6. Available: http://www.acm.org/crossroads/xrds6-2/xml2b.html.

Baravalle, A., Lanfranchi, V. and Gribaudo, M. (2003) 'Using SVG and XSLT for graphic representation'. Scalable Vector Graphics (SVG) Open 2003 Conference. Vancouver, Canada.

Bradin, M. (2002) 'XML Improves Localisation Projects'. Multilingual Computing & Technology 11, 2

Dewar, R. and Olson, P. (2007) 'Human Factors in Traffic Safety. Lawyers & Judges Publishing Company'. Inc., Tucson, Arizona, 719.

Draxler, C., Bengler, K. and Olaverri Monreal, C. (2001) 'Speaking While Driving-Preliminary Results on Spellings in the German SpeechDat-Car Database'. In: Eurospeech 2001 Scandinavia.

Eckstein, R. and Casabianca, M (2001) XML: Kurz& gut, O'Reilly Germany.

Esselink, B. (2000) A practical guide to localization, John Benjamins.

Goodman, M., Tijerina, L., Bents, F. and Wierwille, W. (1999) 'Using cellular telephones in vehicles: Safe or unsafe?'. Transportation Human Factors, 1, 3-42.

Green, P., Hoekstra, E. and Williams, M. (1993) 'On-the-road tests of driver interfaces: Examination of a route guidance system and a car phone'. Technical Report UMTRI-93-35, Ann Arbor, MI: The University of Michigan Transportation Research Institute.

Gueye, B., Rigaux, P. and Spyratos, N. (2004) 'Taxonomy-based annotation of XML documents: Application to elearning resources'. Lecture Notes in Computer Science, 33-42.

Hayes, B., Kurokawa, K. and Wierwille, W. (1989) 'Age-related decrements in automobile instrument panel task performance'. Human Factors and Ergonomics Society Annual Meeting Proceedings, AGING. Human Factors and Ergonomics Society.

Heimgärtner, R. (2007) 'Towards Cultural Adaptability in Driver Information and -Assistance Systems'. Usability and Internationalization. Global and Local User Interfaces. Beijing, China: Springer.

Hudson, R. and Hall, P. (1997) Software without Frontiers: A Multi-Platform, Multi-Cultural, Multi-National Approach, J.Wileys&Sons, London.

Lee, J., Caven, B., Haake, S. and Brown, T. (2001) 'Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers' attention to the roadway'. Human Factors, 43, 631.

Lee, J. and Kantowitz, B. (2005) 'Network analysis of information flows to integrate in-vehicle information systems'. International Journal of Vehicle Information and Communication Systems, 1, 24-43.

Lee, Y., Lee, J. and Ng Boyle, L. (2007) 'Visual attention in driving: The effects of cognitive load and visual disruption'. Human Factors, 49, 721.

Lobin, H. (ed.) (2000) Informationsmodellierung in XML und SGML, Berlin, Heidelberg: Springer- Verlag (2. Aufl. 2001).

Musale, S. (2001) 'Localizing for Mobile Devices: A Primer'.Available:http://www.lisa.org/Globalization-for-Wi.504.0.html#c297.

Neven, F. and Van den Bussche, J. (2002) 'Expressiveness of structured document query languages based on attribute grammars'. Journal of the ACM (JACM), 49, 56-100.

Nussbaumer, A. and Mistlbacher, A. (eds.) (2002) XML Gepackt mitp: Bonn: Verlag.

Olaverri Monreal, C. (2006) Lokalisierung und Internationalisierung von Fahrerinformationssystemen, München: Ludwig-Maximilians-Universität.

Pérez González, L. (2009) 'Audiovisual Translation'. Encyclopedia of Translation Studies. M. Baker and G. Saldanha (eds) ed. London: Routledge.

Rakauskas, M., Gugerty, L. and Ward, N. (2004) 'Effects of naturalistic cell phone conversations on driving performance'. Journal of Safety Research, 35, 453-464.

Rößger, P. (2003) 'Cross Cultural Differences in Human Machine Interfaces of Driver-Information-Systems and how to Cope with them from the Software Side'. ITSA Meeting. Minneapolis, MN.

Rößger, P. (2004) 'Interkulturelle Unterschiede bei der Usability von Fahrer-Informations-Systemen Grundlagen, Methoden und Ergebnisse. '. Entwerfen und Gestalten. Steffens, C. .

Savourel, Y. (2002) 'XML Technologies and the Localisation Process'. MultiLingual Computing & Technology Nr. 35 11 (7).

Sperberg-McQueen, M., Huitfeld, C. and Renear, A. (2000) 'Meaning and interpretation of markup'. Markup Languages 2/3, 215-234.

Spona, H. (2001) Das Einsteigerseminar SVG-Webgrafiken mit XML, KG, Landsberg: Verlag moderne Industrie Buch AG & Co.

Taylor, D. (ed.) (1992) Global Software: Developing Applications for the International Market, New York: Springer.

Tijerina, L., Johnston, S., Parmer, E., Winterbottom, M. and Goodman, M. (2000) 'Driver distraction with wireless telecommunications and route guidance systems'. Report No. DOT HS, 809, 069.

Tijerina, L., Kiger, S., Rockwell, T. and Tornow, C. (1995) 'Workload assessment of in-cab text message system and cellular phone use by heavy vehicle drivers on the road'. Human Factors and Ergonomics Society.

Tuffley, D. (2003) 'Optimizing a Software Internationalisation Curriculum. Designing for Global Markets'. Designing for Global Markets 4, Products & Systems Internationalisation, Inc. Austin, Texas.

Van Otegem, M. (2009) 'Sams teach yourself XSLT in 21 days'. European Journal of Control, 15.

Whitehead, P., Friedman-Hill, E. and Vander Veer, E. (2002) Java and XML: your visual blueprint for creating Java-enhanced Web programs, Third Avenue New York, NY 10022: Wiley Publishing, Inc. 909.

Wierwille, W. and Tijerina, L. (1996) 'An analysis of driving accident narratives as a means of determining problems caused by in-vehicle visual allocation and visual workload'. Vision in Vehicles V. Amsterdam, Netherlands: Elsevier Science Publications.

Yu, Y., Lu, J., Mylopoulos, J., Sun, W., Xue, J. and D'Hollander, E. (2005) 'Making XML document markup international'. Software Practice and Experience [Online], 35. Available: www.interscience.wiley.com [Accessed 18 October 2004].

Web Internationalisation strategies and translation quality: researching the case of "international" Spanish[1]

Miguel A. Jiménez-Crespo,
Department of Spanish and Portuguese
Rutgers, The State University of New Jersey
105 George St
New Brunswick,
NJ, 08901
USA

http://translation.rutgers.edu/
miguelji@rci.rutgers.edu

Abstract

The goal of the Localisation Industry is to produce sites that "look like they have been developed in-country" (LISA 2004, p. 11), even when due to cost-effectiveness considerations international versions are frequently released for languages with multiple locales. Thus, the industry's discourse links quality in localisations to texts that are received as local productions, while its internationalisation strategies strive to erase certain dialectal and cultural differences. This paper researches the strategies applied by multinational corporations when dealing with neutral international version of localised websites. After a theoretical review of the interrelated notions of internationalisation, quality, neutral language version and localisation level, the empirical study researches the case of Spanish, one of the languages with the greater number of different locales. Following a corpus-based approach, the Spanish localisation strategies of the 600 largest US companies are analyzed. Detailed statistics for each model and locale are provided. In a second stage, a longitudinal study is presented that contrasts data collected in 2006 with the localisation strategies observed in 2009. The results show that multinational companies apply different strategies for the European and Latin American markets, while the largest US companies have gradually increased the Spanish localised versions of their websites. Additionally, the fuzzy US Spanish locale continues to grow despite the fact that this locale is not still internationally recognized and standardized.

Keywords: *website localisation, internationalisation, neutral language, translation quality, corporate websites, corpus-based translation studies.*

1. Introduction

DURING recent years, multinational companies have embraced the Internet as their main communicative platform to reach an ever-expanding global market. Globally, the number of Internet users has grown by 380% in the last decade (Internet World Statistics 2009), and this means that corporate websites have ubiquitously become the main communication instrument between companies and their customers (Yunker 2003; Kennedy and Shepherd 2005). It was earlier on in the development of the Internet that companies recognized the users' preference to access web content in the native languages (Yunker 2003) and since then, multinational corporations have been increasing the number of localised versions of their websites. In order to cope with this increasing demand, companies constantly need to develop and implement different localisation strategies.

This paper researches one of the main strategies in the industry: creating an *international* version for a language that is spoken in different countries and cultural regions. The point of departure for this study is the existence of two seemingly contradictory tendencies in the discourse of the Localisation Industry. On the one hand, the goal for the localisation process is to release websites with "the look and feel of locally made products" (LISA 2003, p. 5), and on the other hand, the goal of the internationalisation stage is to develop products that are language and culture independent. In a sense,

[1] A preliminary abridged version of this study was presented at the LISA Berkely Forum on Globalization, Bekeley University, August, 3rd-4th 2009.

there are two tendencies that run in opposite directions, the former focused on user-based quality approaches, with the latter centered on localisation as a time and resource-constrained activity that depends on ROI decisions. This paper represents a first step in a wider study of the the impact of these contradictory tendencies on the quality and usability of localised websites. These internationalisation strategies for languages with different locales are studied through a representative analysis of *neutral* Spanish localisations. This language was chosen because it has the highest number of different standardized locales, that is, the combination of a language and a specific sociocultural region where it is spoken.

As a first step towards understating the potential impact of internationalisation strategies on quality, it is necessary first and foremost to establish a theoretical base. The following sections review several concepts and how they interrelate, such as internationalisation, localisation models and levels, localisation quality, and finally, a discussion on whether a *neutral* or *international Spanish* variety exists.

After this theoretical review, the empirical study will analyze and compare industry strategies and trends, as well as contrast 2009 data with a previous corpus study in which data were compiled in 2006 (Jiménez-Crespo 2008a).

2. Defining the relationship between internationalisation and quality: from process to product

The global localisation process has been normally known by the acronym GILT, Globalization, Internationalisation, Localisation and Translation (Dunne 2006). While globalization mostly deals with organizational and business aspects, internationalisation is mainly understood as a technical stage in which a product is enabled for localisation (LISA 2007, p. 17). The goal of this stage is to guarantee to the highest possible degree that functional and development aspects are not culture-specific, so as to not pose any problems while localisation and translation take place. The Localisation Industry Standards Association also defines the goal of this process as:

> [A]bstracting the functionality of a product away from any particular language so that language support can be added back in simply, without worry that language-specific features will pose a problem when the product is localised (LISA 2004, p. 14).

The rationale behind this technical process is that products should be developed in a culturally neutral form, something that is theoretically impossible as there are culture and language specific variables that cannot be fully controlled (House 2001; Shreve 2006). Normally, this process requires a collaborative effort between developers and translators in order to produce the most neutral possible source website. This internationalisation process at a development stage should not be confused with the internationalisation of languages with multiple regional varieties. This effort to establish a neutral version of the multinational language entails a completely different linguistic and cultural process. The development of a product using an international language is normally carried out at the translation stage of the global GILT process. It is also performed by terminologists and translators, while the technical cycle is informed by translators and implemented by developers. Figure 1 shows the interrelated nature of this global cycle.

The neutralization or language internationalisation is therefore carried out after the technical *internationalisation* stage, once the source website is localised into all requested locales[2]. Nevertheless, when a language-neutral strategy is adopted, the industry implicitly associates one international language with a supranational culture, i.e. the Spanish culture in the case of Spanish, even when the underlying principle behind the adoption of locales was the need to separate languages from sociocultural regions and their specificities. In this context, the industry normally applies a fuzzy definition of language in which linguistic aspects can be separated from cultures, even when all languages are culturally situated and they form a unitary whole (Bassnet and Lefevre 1990). The differences between locales exist not only at the linguistic level, but many other levels are also culture dependent, such as the pragmatic or discursive ones. Locales from a single language can thus differ in certain culture-dependent textual aspects, such as digital genre conventions that can differ from from culture to culture (Nord 1997; Jiménez-Crespo 2008c, 2009a). These digital genres, such as corporate or social network sites, are highly conventionalized communicative instruments that represent the functions and goals involved in a particular web interaction as well as the purposes of the participants in them (Hatim and Mason 1990, p. 69). One of the main applications of genre theory to

translation and localisation is that contrastively, textual structure, phraseology and terminology associated with each constituent communicative block or section in a genre, such as contact forms or the terms in a privacy page, can show differences between source and target cultures (Jiménez-Crespo 2008c). The same can be said of languages that do not fully share the same culture, such as Spanish in Mexico and Spain or English in the UK and South Africa. Thus, language-neutral strategies are mostly concerned with linguistic aspects even when discursive, rhetorical and textual aspects dependent on genres can also be differently conventionalized.

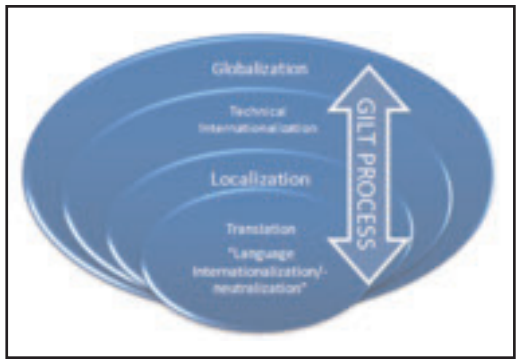


Fig. 1. GILT process. Adapted from Dunne (2006).

Conventions play a vital role in web usability, as it has been empirically shown that websites that do not follow conventions lead to lower usability, comprehension, recall, etc. (Nielsen 2000; Nielsen and Tahir 2002; Nielsen and Loranger 2006; Vaughan and Dillon 2006). In part, this is due to the fact that websites are instrumental texts that are not necessarily written to be read linearly; quite on the contrary, websites are normally scanned in order to identify the information that the user is looking for (Nielsen and Loranger 2006). Thus, when deviating from cultural and linguistic conventions in order to internationalise any website, the subsequent potential impact on web usability should always be taken into consideration.

Nevertheless, producing a neutral language version does not automatically mean that certain culture-specific adjustments will not be made, as this depends on additional localisation decisions regarding the level of cultural adaptation or *localisation level.*

2.1. The degree of customisation or localisation level and its relationship to localisation models.

In localisation, which is a business process with finite time, human and economic resources, the degree of customisation always depends on the importance of the local market for the business activity of the company. Normally, the decisions about the level of customisation for different countries that share the same language always depend on Return on Investment (ROI). These economic and business decisions result in a localisation level, a concept that has been defined as:

> The amount of translation and customisation necessary to create different language editions. The levels, which are determined by balancing risk and return, range from translating nothing to shipping a completely translated product with customized features (Microsoft 2003, p. 15).

In the first publication that mentioned the notion of localisation level, Brooks (2000, pp. 49-50) described the practices of Microsoft, where the software products were localised according to three distinct levels:

> 1. Enabled products: Those in which users can write and use their own language and scripts, but the software and the accompanying help and guides appear in a different language.
>
> 2. Localised products: Those in which the user interface and all help files are localised, but some language-specific tools such as spell checkers and dictionaries are not available.
>
> 3. Adapted products: Those in which all linguistic tools, functionalities and content are fully adapted to the target language/locale.

In the case of web localisation, Yunker (2003) and Singh and Pereira (2005, pp. 10-15) have proposed a different categorization for web localisation levels. The latter is the most detailed proposal:

> 1. Standardized websites: In which a multi-national company simply offers a site in one language for all countries/markets.
>
> 2. Semi-localised websites: In which the only locale/specific content is a contact page in the target language with information about local

[2] Some larger companies, such as Microsoft, develop their "neutral" language versions both during a preliminary stage in their terminology departments, as well as in a post-localisation stage using QA feedback and crowdsourcing through their linguistic portals, http://www.microsoft.com/language/mtcf/mtcf_default.aspx.

branches, contacts, etc.

3. Localised websites: In which most content and pages are localised, but the original functionalities and back-end are not modified.

4. Extensively localised websites: In which there is a global localisation and all content and site structure/ functionalities are fully adapted to the target locale.

5. Culturally adapted websites: This is the most advanced level of localisation, the one that the authors advocate, and in which there is a total immersion in the target locale. Sites are adapted to the levels of cultural descriptions proposed by Hofsteade (1991): perception, symbolism and behavior.

It can be observed that the former proposal does not mention explicitly the existence of international versions for specific languages such as French or Spanish, while the latter refers mostly to cultures, that is, the combination of a language and a sociocultural region or locale. It is also interesting to note that, in the latter description, the lower levels tend to mention *languages*, while as the localisation level increases the terms *locale* and *culture* are preferred. Thus, an analysis of these models suggests that lower levels tend to deal with languages, while higher levels require the more in-depth linguistic and cultural adaptation to specific locales.

Additionally, the industry implements two distinct models in web localisation depending on whether the website for a specific market is the result of a localisation process or an original production. These are the so-called *centralized* vs. *decentralized* models (Yunker 2003, p. 128; O´Hagan and Ashworth 2003, p. 74). In the former, a company produces an international site that is localised and customized for any targeted locale from a centralized location, as it is the case of Google or Microsoft. All versions that result from this model can be considered the result of a localisation and translation process in which linguists are involved. In the "decentralized model", companies such as Pfizer develop a "shell" and the resulting websites for Spain, Mexico or Argentina are the product of a local production, and therefore, the resulting web content is not the product of a translation process but rather natively produced content in each locale.

It is obvious that the importance of the market and decisions taken in regards to the localisation level will have some impact on one of the most important goals of the localisation process: producing quality localised websites. The notion of quality is a highly debated and researched notion in Translation Studies, where more and more researchers are trying to create bridges between research and the professional world. In order to understand the potential impact of these strategies on quality, the following section reviews how the industry conceptualizes and operationalises this concept.

2.2. Quality in the Localisation Industry

The localisation industry has implemented quality standards and metrics directly based on international norms, such as ISO, TQM or the European Union EN-15038 quality standard. In them, quality is generally defined as the capacity to comply with a set of parameters pre-defined by the customer. For example, the ISO 9000 defines quality as: "the totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied needs" (ISO 1993). With a very similar perspective, TQM (Total Quality Management) defines quality as "fully satisfying agreed customer requirements". Nevertheless, it would be theoretically and methodologically impossible to predefine the notion of "quality" in all translated texts: for this reason, common definitions of quality usually focus on procedural aspects (as above) as opposed to establishing what could be considered a "quality" translation. Basically, such definitions govern procedures for achieving quality, rather than providing normative statements about what constitutes quality (Martínez Melis and Hurtado 2001, p. 274). Additionally, they are generically process-oriented instead of product-oriented (Wright 2006, p. 256; Corpas 2006). Again, quality in the industry is understood as a construct that can be controlled in the process, irrespective of characteristics that are reflected in the product.

If published literature by the Localisation Industry is thus analyzed, a quality localised product equals a website with (1) a limited number of previously defined errors (Bass 2006), which (2) looks like a natively produced website (LISA 2004), and (3) whose functionality is not compromised. An additional item should be also added, (4) the ability to effectively satisfy web users' implied needs and attract repeated visits, an objective related to web usability (Nielsen and Lorangar 2006). In order to achieve this goal, the industry uses time-constrained evaluation QA processes that are carried out by one

or more evaluators. As with many translation evaluation processes, these evaluators might lack the necessary theoretical framework in order to separate their own subjective judgments from more objective interrater criteria (House 2001; Hönig 1998, p. 14). Additionally, the notion of quality is understood as the relative absence of errors (Bass 2006), despite the fact that current research in translation evaluation has shown that relying solely on error based approaches is insufficient to fully assess quality (Jiménez-Crespo 2009b).

Moreover, if the objective of the industry is to produce texts that are received as if they had "been developed in-country" (LISA 2004, p. 11), localisation strategies should in principle guarantee that websites serve their purpose efficiently and share whichever characteristics are conventional in each locale (Nielsen 2000; Nielsen and Tahir 2002; Nielsen and Loranger 2006) . It is therefore logical to argue that, apart from eradicating any transfer and language errors in the target language, a quality localised website should share the characteristics of a similar locally made website. In order to accomplish this goal, compliance with existing cultural and linguistic conventions in natively produced texts becomes an essential aspect of quality (Jiménez-Crespo 2009a; Nielsen and Tahir 2002). This issue then goes beyond error identification and requires full active competence in the genres in question (Gamero 2001), that is, translator-evaluators should have the ability to produce these highly communicative instruments. Additionally, quality evaluation procedures should be able to guarantee that underlying or "hidden" conventions from the language/culture of development can be identified and controlled (House 2001). Therefore, in terms of quality analysis, the questions that these issues pose are many, and future empirical investigations will focus on them: does the existence of *internationalised* sites mean that there is a global *internationalised* Spanish corporate site genre? Can corporate websites be similar in all 20 Spanish locales? Is it possible to produce a localised version that all Spanish-speaking users can perceive as directly addressed to them? And finally, can textual and cultural aspects of an internationalised or *non-culture specific* site look the same or be equally received in twenty different countries? In order to establish a framework to research these questions, the methodology and principles of corpus-based translation studies can be extremely useful.

2.3. Localised texts as a distinct third code

In the light of over two decades of corpus-based translation studies (Baker 1993;Laviosa 2002; Olohan 2004), it can be stated that, even when users might perceive a translation as a natively produced text, translated products do possess different characteristics from natively produced texts (Baker 1993). This is due to the fact that translation is "a communicative event which is shaped by its own goals, pressures and context of production" (Baker 1996, p.175), upon which several specific constraints operate, which can be social, cultural, ideological, technological or cognitive in nature (Baker 1999, p. 285). It is therefore understood that translated websites are the result of a distinct process that results in texts with differentiated characteristics from natively produced ones. Following several scholars that have coined terms for this result, such as *third code* (Frawley 1984), *the third language* (Duff 1981) or *hybrid language* (Trosborg 1997), localised websites can be said to show a specific *language of localisation* (Jiménez-Crespo 2008a; 2009a). This implies that the language of localised websites might be intrinsically different from that of natively produced ones. This is similar to the language of dubbing or *dubbese* that can be found in dubbed movies worldwide (Romero Fresco 2006). Moreover, internationalisation strategies and their push to provide international language versions might represent an additional layer of distinction between original and localised sites; a localised website exclusive for Argentina could have different features from a site originally produced in this country and addressed to Argentinean users, and clearly, it will show even more distinct features if it is a global international Spanish localised version for the Argentinean users. Obviously, compromises need to be made in order to provide this neutral version. Nevertheless, it should be mentioned that different countries and cultures possess diverse degrees of tolerance towards foreignization in translation (Venuti 1995). One of the possible means by which translators can identify the most frequent and conventional uses of languages in each specific locale is through corpus analysis (Jiménez-Crespo 2009b; 2008a). Corpus studies based on carefully selected representative collections of web genres in each locale can also lead to descriptive style guides where conventional items in a specific digital genre can be identified (Jiménez-Crespo 2009b). Additionally, there are two more possible ways to analyze users' quality expectations for internationalised sites, crowdsourcing localisation such as the Facebook model (O´Hagan 2009), and

web usability research (Adkisson 2003; Nielsen 2000; Nielsen and Loranger 2006).

As an example, the conventional lexical unit that corresponds to contact us in Spain is *contacto* (2009a), while in Argentina and Chile it is appropriate to use *contáctese con la empresa.* The use of the Spanish reflexive form of this verb would be identified as a syntactical error by users in Spain, thus affecting any potential Castilian Spanish user´s perception of quality. A website explicitly directed at users in Spain that would use the latter term might be perceived as a low-quality site, mostly due to the low tolerance for grammatical errors in websites (Jeney 2007). Nevertheless, the conventional form in Spain, *contacto*, would be accepted as a natural and conventional term in all these Spanish speaking regions. Thus, most decisions taken in internationalised language sites require extensive research in order to guarantee the best possible localisation.

2.4. In search of *international* Spanish

Spanish is the official language in 21 countries and it currently has over 400 million native speakers. It is the third most used language in the Internet after English and Chinese, with 137 million users in 2009. This represents around 8% of the global Internet population (Interent World Statistics 2009). Spanish is also extremely suited for this study as it is the language with the most number of standardized locales. According to Microsoft's locale registry, there are currently 20 Spanish locales, followed by 18 English ones, 16 Arabic and 15 French. The norm that is used by all Spanish speakers is referred to as *international neutral Spanish* (Castro 2001), and this refers to the international norm that is used in specialized and semi-specialized domains, such as science and technology. This is also referred to as *castellano general, español común, español internacional or español estándar,* and García Izquierdo (2006, p. 152) suggests that the most common denomination for this concept is *español neutro or simply neutral* Spanish.

The origins of this concept can be traced back to either the US film and cartoon producers in Puerto Rico in the 50´s, or the Mexican film producers of the 60´s (García Izquierdo 2006, 2009). That is, the first failed attempts at establishing a common neutral Spanish version were due to economic issues related to establishing neutral versions for audiovisual products. These initial attempts in the film industry are gaining a new momentum among the web and software producers of the 21st century. This issue is also being researched from new perspectives, such as funsubs and crowdsourcing in order to record which is the most widely accepted terminology, phraseology and other linguistic and cultural conventions (O´Hagan 2009).

Currently, the main vehicles for the standardization of the non-specialized Spanish norm are the Internet and media productions. On the Internet, users around the world can constantly access contents in all other Spanish dialectal varieties, including the emerging US Spanish variety and different types of neutral versions. It should be mentioned that the Spanish norm is mostly different at the phonological and lexical level, with no differences whatsoever at the typographic and spelling levels (García Izquierdo 2006, 2009). The differences in the Spanish varieties that can have an impact on web content are therefore mostly terminological and phraseological (such as the terms *inversores-inversionistas*), morphosyntactical (such as different forms in the 3M site for the term contact us: *contácte* a 3M [Honduras], *contáctese con 3M* [Perú, Bolivia, etc.], *contactar con* 3M [Spain]), or stylistic (*descubra cómo* [discover how], *sepa cómo* [learn how]). The main problem in establishing a *neutral* variety for web localisation process relies on the fact that websites are *complex digital genres*, that is, a single website or webpage incorporates different textual types and registers. As an example, any homepage will incorporate *interface* texts (Price and Price 2002) such as navigation menus and search options, advertisements, excerpts from other parts of the site such as technical or legal descriptions, etc. Normally, specialized domains such as technical descriptions are easier to develop and translate in neutral Spanish, while others in a general language such as marketing texts show greater differences and therefore pose greater difficulties in the neutralization process (García Izquierdo 2009).

This theoretical review has attempted to define the interelated and inseparable concepts in this investigation: internationalisation, international Spanish, quality and usability. The next section presents the initial empirical investigation in the wider study into the impact of internationalisation strategies on localisation quality and usability. The underlying rationale behind the wider study is therefore that sites subject to an internationalisation process to neutralize a multilocale language will enhance those characteristics that differentiate localised products from natively produced ones. As a

first step towards empirically testing this hypothesis, this specific study into current strategies in the industry is required.

3. Empirical study

This particular study focuses on current localisation strategies of US multinational companies when dealing with languages that are shared with a number of countries or locales. The starting point is previous studies by the same author into the distinct characteristics of localised sites if compared with natively produced sites (Jiménez-Crespo 2008a, 2009a, 2009c). These studies provide the analysis of a corpus of localised websites from US corporations compiled in 2006. Among other findings, it was observed that, even though users might not identify the language of a site as a translation, many of the linguistic and cultural features that appear in these sites will never be spontaneously produced by a target locale web developer. Following a corpus-based methodology, the current analysis will concentrate on researching the Spanish localisation strategies of the largest US companies in 2009.

3.1. Methodology

As previously mentioned, the methodology for this study was based on a previous corpus-based study that analyzed the websites of the 600 largest US companies according to the Forbes List. The corpus data compilation took place in 2006, and the same list was used in order to contrast the current localisation strategies in 2009. In the latter analysis, only the first 100 companies were studied. From the original 2006 Forbes list, four companies had disappeared, three of them due to mergers and one to bankruptcy. The next four companies in the list with a Spanish version of their website were added to the list.

A detailed analysis of the localisation strategies for the Spanish speaking locales was carried out in the 2009 list, and the process can be summarized as follows: (1) recording the number of Spanish locales that were used in each global website, (2) analyzing whether the site was a localisation exclusively for one locale or what kind of locale groupings were made in their Spanish language websites; the analysis was carried out on the terminology in navigation menus, web banners and visual elements, (3) whether the localised version was presented as a custom local production for a specific locale or, on the contrary, the localised website was presented as an international Spanish site. The analytical process to identify whether a website was an original production or a localisation followed the 10 step criteria previously developed (Jiménez-Crespo 2008a, pp. 233-234), such as the existence of an English and a Spanish version, the existence of frames in another languages, analysing potential translation errors, and most importantly, analysing the source code to identify whether the comments (<!), image names etc. are in Spanish. This last step indicates that the website as a whole was originally developed in Spanish, as normally localised sites maintain developers´ comments in the source language.

In a second stage, the results from the previous corpus analysis compiled in 2006 are contrasted with the current results in order to identify current industry trends. The 2006 and 2009 data will be compared both for the percentage of companies offering localised versions of their sites and the relationship between company size and the adopted localisation model.

4. Results and discussion

In the first place, the websites of the 100 largest US companies with Spanish localised versions were analyzed in order to identify the number of locales or markets targeted by the industry. Figure 2 presents a summary of the percentages of Spanish locales that the websites explicitly indicated in their localisations, that is, that a website was directly aimed at at this specific country. This is done by means of indicating the country/language combination (e.g. Español-Honduras), the country site (e.g. 3M Bolivia), or a map with the specific region. Three additional locales were added as the websites showed that the versions were addressing the *Caribbean region, Latin America* and *Central America*. Thus, the industry did not exclusively use the standardized locale IDs consisting of the combination of country and language, but rather, geographical denominations other than countries were used in lieu of standardized country codes. Additionally, the locale Spanish-US was added given that many of these corporations present a Spanish site for their Spanish-speaking customers in this country. It can be clearly observed that the largest targets for Spanish localisation are: Spain (42%), Mexico (32%), the United States (27%) and Argentina (27%). This analysis also indicates that the top 100 US companies directly present their localisations as country-oriented, rather than language-oriented, even when many of the localised versions might be shared by many of those countries. Obviously, the strategy of those companies is not to

produce a Spanish localised version for each of those countries, but rather, to combine and group the countries in different ways depending on their market shares.

Figure 2. Percentage of Spanish locales targeted by the largest US corporations.

The next step in the analysis was to analyze what strategies the companies were implementing to deal with the multiple Spanish localised versions of their sites. Overall, four main strategies were identified:

1) A *decentralized* model (Yunker 2003) in which each website is directly developed in each locale and it entailed an original local production maintaining the corporate visual identity.

2) A *centralized* model (Yunker 2003) in which different localisations into Spanish are provided for different locales or groups of locales.

3) A centralized model in which a single localised version is presented as a local production for each targeted locale, that is, even when the localised version is the same. Users can select their own country from the locale selection option and the site is presented to them as localisation exclusively for that country.

4) A single international website for all locales presented as an international site in Spanish.

The most interesting discussion in this regard, a hypothesis that will be empirically tested in subsequent studies, is that each model adopted might entail different implications in terms of users´ quality perception. In the first model, users interact with a local production, the most expensive option, and the one that represents the highest possible level of localisation to a target locale. The next possibility, producing locale-specific Spanish localised versions, also responds to the need to adapt the content to the specificities of each country or region. The third and fourth models, the most widely used due to ROI and economic issues, presents different advantages and drawbacks. A user faced with an international version of a Spanish site is aware that the company is not directly focused on this market and therefore, is not addressing him/her directly, but at the same time, the user will be more tolerant towards any linguistic or cultural item that the user associates with a different dialectal variety or Spanish-speaking country. Therefore, the potential impact on quality of other Spanish variants should be lower. On the other hand, a localised site directly aimed at users in Argentina makes users believe that the company is committed to this market, but nevertheless, the impact on the user´s perception of quality of any non-Argentinean cultural or linguistic use might be high, as the user can feel deceived. In part, this can be due to the fact that in certain cases some dialectal variants are perceived as linguistic errors.This is comparable to a website developed in the UK with British spelling and colloquialisms that would be implicitely addressed at US users as a local production.

A detailed analysis of all localised versions was carried out to identify which of these four models were used for each company. Figure 3 shows an analysis of the adoption of these four different strategies. As the European Spanish market represents the largest share, it is separated from the rest of the locales.

It can be observed that the industry favors different localisation strategies for the European and Latin American markets. In the European Spanish locale, 30% of the companies release localised versions exclusively produced in this dialectal variety, while the most frequent strategy in the Latin American market is a single localised version that is, nevertheless, presented as an exclusive localisation for each locale (17%). Both the European and Latin

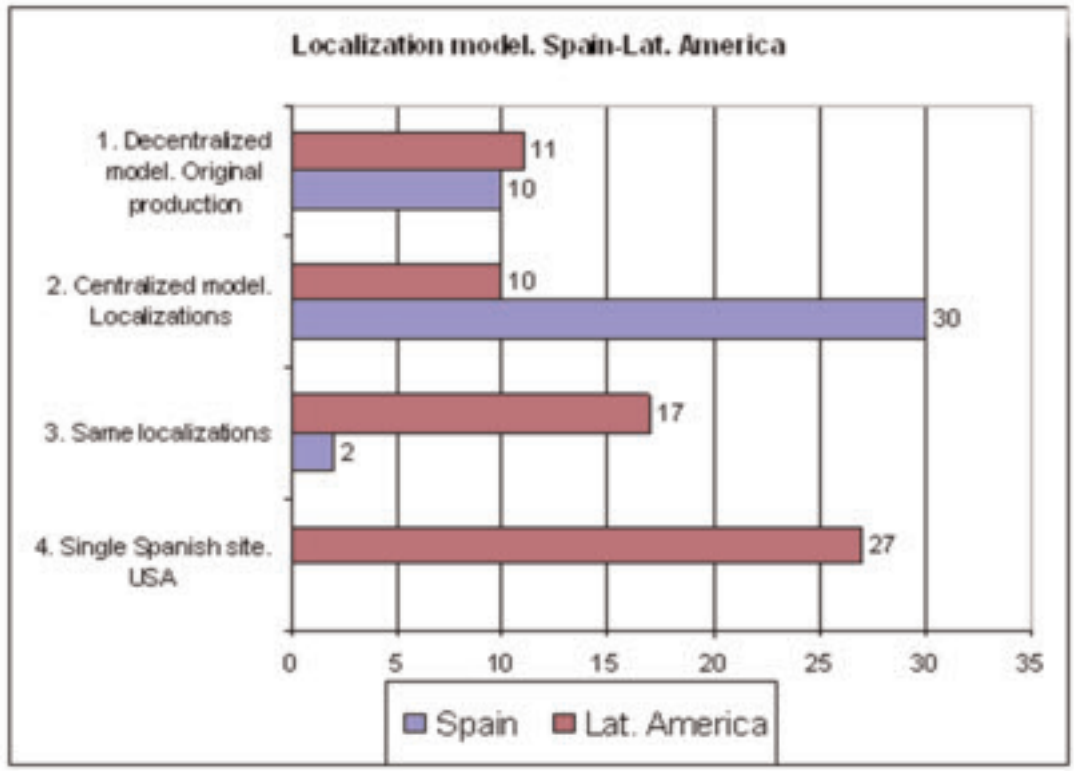


Figure 3. Localisation models for the Spanish-speaking locales in the largest US companies.

American markets present very similar percentages of websites that are not the result of a localisation process, but rather an original production for each locale: 10% for European Spanish and 11% for Latin American countries. It should also be mentioned that companies very rarely implement a single localised version for all Spanish-Speaking locales and then present it as a local production in Spain (2%). Finally, most international Spanish sites are those produced for the US market, and officially, this tends to be their global version for any other Spanish-speaking customer in the world.

4.1.Spanish localisation trends from 2006 to 2009

The last two analyses involve a comparison of the localisation trends between the results obtained in 2006 and 2009. First of all, the last analysis concentrates on the localisation trends in terms of companies offering Spanish localisation for their sites. The analysis was carried out using the data for the first 100 companies in the 2006 analysis and the 2009 list. Figure 4 offers a glimpse of how the number of Spanish localisations is constantly growing.

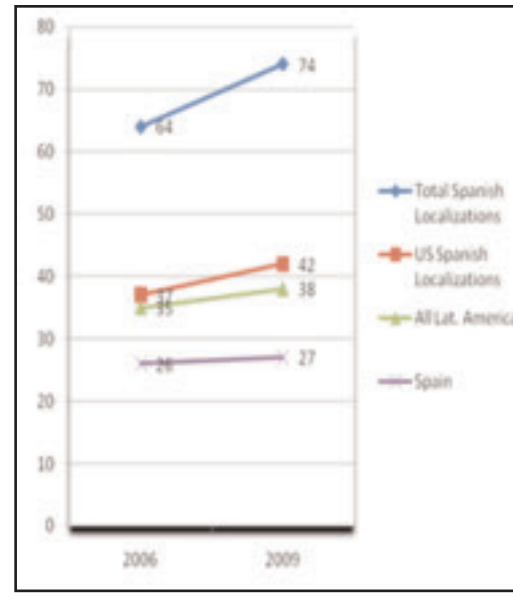


Figure 4. 2006/2009 localisation trends for US companies offering Spanish localised versions of their sites.

Globally, the number of companies offering a localised version for any Spanish locale has grown by 11.56% in these three years. The data for most locales has grown in very similar terms. For the European Spanish locale, there is an increase from 37 to 42 companies offering a site in Spanish, an 11.35% growth rate. Similarly, the increase in localisations for Latin American locales also jumped by 10.85%. Finally, the number of companies offering a site in Spanish for US customers climbed by 10.38%. largest companies in 2006 were contrasted with the 100 largest US companies in the 2009 list. The most frequent locale, European Spanish, was used in this analysis.

The contrastive analysis shown in Figure 5 shows that, despite the 11.56% increase from 2006 to 2009, the possibility of offering a Spanish localisation is directly proportional to company size. Overall, 74% of the largest US companies offer a localised version

Fig.5. Screen capture from the Spanish localisation for the Home Depot site explaining that this localisation has disappeared.

Nevertheless, it should be mentioned that the real increase was larger, but it was reduced by the fact that some companies, such as Home Depot, decided to eliminate their localised versions and some other companies disappeared or merged. Interestingly, there might be a relationship between users not relating to some corporate US sites in Spanish and the lack of existence of a clearly established US Spanish variety, thus impacting on the site usability or the user's lack of response to the site.

Finally, the percentage of companies with a localisation into Spanish was contrasted in order to observe whether the possibility of offering a localisation decreases or increases according to company size. It is expected that, despite the observed increase in localisations from 2006 to 2009, the larger the company would be, the more likely it would be that it would localise its website into Spanish. For this purpose, the results for the 600 into any Spanish locale, while the frequency is reduced to 41.38% for the group of the 600 largest companies. There is also an interesting correlation between company size and the possibility of offering an original production as the local website instead of a localised version. For European Spanish, the possibility of offering an original production instead of a localisation is 2.98 times greater for the 100 largest companies if contrasted with the 600 group. In the case of localised versions of sites, the ratio is 1.71 times higher. Thus, it can be clearly observed that the most expensive localisation model, to develop an in-country website, is directly related to company size. As previously mentioned, this could be the model with the higher levels of potential quality from a linguistic standpoint.

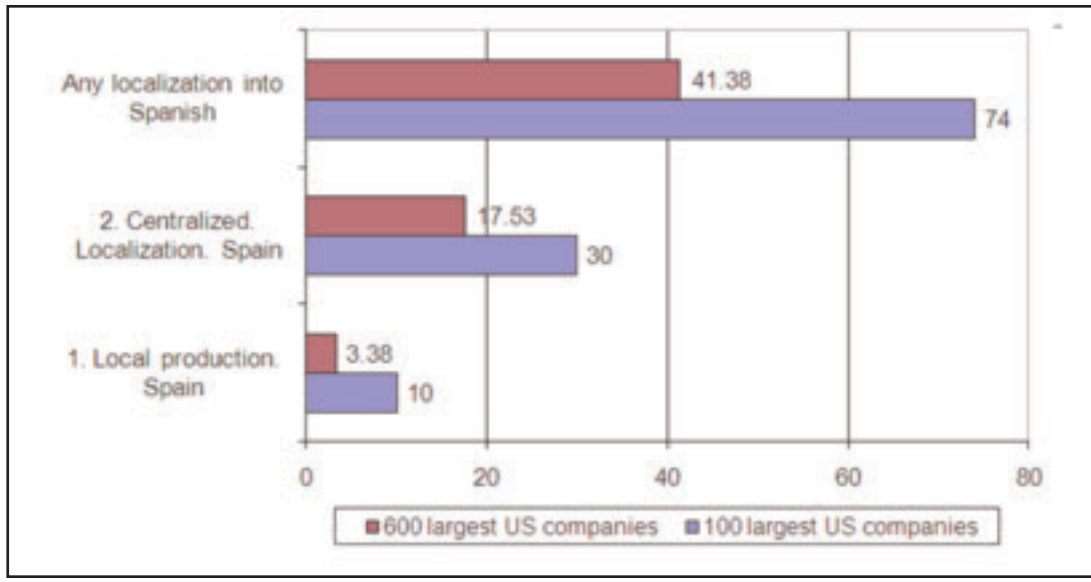


Figure 6. Contrastive analysis of localisation model and company size.

5. Conclusions

Translation quality is a multidimensional concept that can be controlled through a number of approaches, each of which presents its possible benefits and drawbacks. The approaches range from current process-oriented approaches based on international standards to user-based ones such as the web crowdsourcing translation quality implemented by Facebook. In all of them, the objective is to produce a website that complies to the user´s implied needs and that adjusts to the linguistic and cultural framework of expectations. Once established that the objective of the industry is to produce websites that users can accept as *locally made* (LISA 2003), the interrelations between internationalisation, quality and *neutral-language* versions has been explored.

In this empirical investigation, four main strategies have been identified, and an analysis of industry trends has been presented. Mainly, it is clear that the industry separates the Spanish and the Latin American markets, and different strategies are adopted in each case. For Spain, the most frequent strategy is to present a localised version of the site for this locale, while in the Latin American market the main strategy is to present a single localised site that nevertheless is presented as a local production for each targeted locale. Additionally, the main market for neutral Spanish is the United States in which 27% for the largest corporations offer a Spanish localisation. It has also been shown that the localisation model adopted is directly related to company size, thus, the larger the company, the higher the possibility of offering an original site for the market targeted; followed by a localisation for each locale and lastly, a single localisation for all targeted locales.

Finally, it has been shown that consistently in the last 3 years, the volume of web-site localisation into Spanish has steadily grown, with an average increase of around 10 - 11%. This is consistent with the unprecedented growth of Internet penetration in the Spanish language market (Internet Wold Statistics 2009). Nevertheless, the opposite tendency has also been observed, mainly in the US market where some Spanish localised versions have disappeared. This can be due to either the lack of budget for localisation or to the lack of interest by the US Spanish speaking population. The disappearance of Spanish localisations could be closely related to the difficulty in establishing a neutral version of the Spanish language for the entire US market, if the *language of localisation* does not comply with the framework of user´s expectations in regards to genre conventions at the lexical, terminological and phraseological levels, users might consciously or subconsciously find that the website is difficult to use and opt for using the English version. This presents an interesting case since many US Spanish users are bilinguals and can use the source English site, while in other locales if the linguistic quality of the site does not match their expectations or is perceived to complicate its use,

there is no other alternative for monolingual customers.

This study represents the first step in a larger study into the relationship between quality, internationalisation and the role of user expectations and the uneven relationship between dominant and receiving cultures in translation (Venuti 1995). Additional user-based studies would be needed in order to empirically assess the impact of neutral Spanish sites and the different models to deal with this issue. It is hoped that this study will be useful to practitioners, translation managers and industry experts alike, given that, despite economic and time constraints, all participants share the same goal of offering the users the highest possible quality in their interaction with their websites.

References

Adkisson, H. (2002) Identifying de-facto standards for E-commerce websites, unpublished M.A. thesis, University of Washington.

Baker, M. (1993) 'Corpus Linguistics and Translation Studies: Implications and Applications', in Baker, M., Francis, M. and Tognini-Bonelli , E. (eds.) Text and Technology: In Honour of John Sinclair, Amsterdam-Philadelphia: John Benjamins, 233-250.

Baker, M. (1996) 'Corpus-based Translation Studies: The Challenges that Lie Ahead', in Somers, H. (ed.) Terminology, LSP and Translation: Studies in Language Engineering in Honour of Juan C. Sager, Amsterdam-Philadelphia: John Benjamins, 175-186.

Baker, M. (1999) 'The Role of Corpora in Investigating the Linguistic Behaviour of Professional Translators', International Journal of Corpus Linguistics, 4, 281-298.

Bass, S. (2006) 'Quality in the Real World', in Dunne, K. (ed.) Perspectives on Localisation, Amsterdam-Philadelphia: John Benjamins, 69-84.

Bassnet S. and Lefevre, A. (1990) Translation, history and culture, London/New York: Pinter Publishers.

Brooks, D. (2000) 'What Price Globalization? Managing Costs at Microsoft', in Sprung, R. C. (ed.) Translating into Success. Cutting-edge strategies for going multilingual in a global age, Amsterdam-Philadelphia: John Benjamins, 42-59.

Castro, X (2001) 'El ciberspanglish, es español comercial y el español neutro en la red', Panace@, 2, 36-41.

Corpas, G. (2006) 'Translation Quality Standards in Europe: An Overview', in Miyares Bermúdez, E. and Ruiz Miyares, L. (eds.) Linguistics in the Twenty First Century. Cambridge: Cambridge Scholars Press/Santiago de Cuba: Centro de Lingüística Aplicada, 47-57.

Duff, A. (1981) The third language: recurrent problems of translation into English, Oxford: Pergamon Press.

Dunne, K. (2006) 'A Copernican Revolution', in Dunne, K. (ed.) Perspectives on Localisation. Amsterdam-Philadelphia: John Benjamins, 1-11.

Frawley, W. (1984) 'Prolegomenon to a theory of translation', in Frawley, W. (ed.) Translation. Literary, linguistic and philosophical perspectives, London-Toronto: Associated University Presses, 159-175.

Gamero Pérez, S. (2001) La traducción de textos técnicos. Barcelona: Ariel.

García Izquierdo, I. (2009) 'El español neutro en los discursosde especialidad: ¿mito, utopía o realidad?', Ikala, revista de lenguaje y cultura, 23, 15-38.

García Izquierdo, I. (2006) 'El español neutro y la traducción de los lenguajes de especialidad', Sendebar, 17, 149-167.

Hatim B. and Mason, M. (1990) Discourse and the Translator. London/New York: Longman.

Hönig, G. (1998) 'Positions, Power and Practice: Functionalist Approaches and Translation Quality Assessment', in Schäffner, C. (ed.) Translation and Quality, Clevedon: Multilingual Matters, 6-34.

Hofstede, G. (1991) Cultures and Organizations: Software of the Mind. London: McGraw-Hill.

House, J. (2001) 'Translation Quality Assessment: Linguistic Description versus Social Evaluation', Meta, 46, 243-257.

House, J. (2001) 'Beyond intervention: universals in translation' Trans-kom, 1, 6-19.

Internet World Statistics, World Internet Users and Population Stats. 2009. http://www.internetworldstats.com/stats.htm

ISO 9000: Quality management information systems: guidelines for enterprises in developing countries (1993). Geneva, Switzerland: ITC, ISO.

Jeney, C. (2007) Writing for the Web: a Practical Guide, Columbus, OH: Pearson Prentice Hall.

Jiménez-Crespo, M.A. (2009a) 'Conventions in localisation: a corpus study of original vs. translated web texts', Jostrans: The Journal of Specialized Translation, 12, 79-102, available http://www.jostrans.org/issue12/art_jimenez.php [accessed 9 Jul 2010]

Jiménez-Crespo, M.A. (2009b) 'The evaluation of pragmatic and functionalist aspects in localisation: towards a holistic approach to Quality Assurance' The Journal of Internationalization and Localisation, 1, 60-93.

Jiménez-Crespo, M.A. (2009c) 'The effect of Translation Memory tools in translated web texts: evidence from a comparative product-based study' Linguistica Antverpiensia, 8, 213-232.

Jiménez-Crespo, M.A. (2008a) El proceso de localización web: estudio contrastivo de un corpus comparable del género sitio web corporativo, Doctoral Dissertation: Universidad de Granada. Available at http://hera.ugr.es/tesisugr/17515324.pdf.

Jiménez-Crespo, M.A. (2008b) 'Caracterización del género 'sitio web corporativo' español: análisis descriptivo con fines traductológicos', in Fernández Sánchez, M. and Muñoz Martín, R. (eds.) Aproximaciones cognitivas cognitivas al estudio de la traducción e interpretación, Granada: Comares, 259-300.

Jiménez-Crespo, M.A. (2008c) 'Web genres in Localisation: a Spanish corpus study', Localisation Focus - The International Journal of Localisation, 6, 4-14.

Kenny, D. (2001) Lexis and Creativity in Translation. A corpus-based study, Manchester: St. Jerome.

Kennedy, A. and Shepherd, M. (2005) 'Automatic Identification of Home Pages on the Web', in Proceeding from the XXXVIII Annual Hawaii International Conference on System Sciences, Maui, Hawaii. Los Alamitos, CA: IEEE-Computer Society, 2005.

Laviosa, S. (2002) Corpus-based Translation Studies, Amsterdam: Rodopi.

LISA, (2003) Localisation Industry Primer, Geneva: The Localisation Industry Standards Association (LISA).

LISA, (2004) Localisation Industry Primer, 2nd Edition. Geneva: The Localisation Industry Standards Association.

LISA (2007) LISA Globalization Industry Primer. Romainmôtier, Switzerland: Localisation Industry Standards Association.

Martínez Melis, N. and Hurtado Albir, A. (2001) 'Assesment in Translation Studies: Research Needs', Meta, 47, 272-287.

Microsoft Corporation (2003) Developing International Software, Redmond: Microsoft Corporation.

Nielsen, J. (2000) Designing Web Usability: the practice of simplicity, Indianapolis: News Riders.

Nielsen, J. and Loranger, H. (2006) Prioritizing Web Usability, Indianapolis: News Riders.

Nielsen, J. and Tahir, M. (2002) Homepage usability: 50 Websites deconstructed, Indianapolis: News Riders.

Nord, C. (1997) Translating as a Purposeful Activity. Functionalist Approaches Explained, Manchester: St. Jerome.

O'Hagan, M. (2009) 'Evolution of User-generated Translation: Fansubs, Translation Hacking and Crowdsourcing', The Journal of Internationalization and Localisation, 1, 94-121.

O´Hagan, M. and Ashworth, D. (2003) Translation-Mediated Communication in a digital World: facing the challenges of Globalization and Localisation, Clevendon, England: Multilingual Matters.

Olohan, M. (2004) Introducing Corpora in Translation Studies, London: Routledge.

Price, J. and Price, L. (2002) Hot text. Web Writing that works, Berkeley, CA: News Riders.

Romero-Fresco, P. (2006) 'The Spanish Dubbese: A Case of (Un)idiomatic Friends', Jostrans, The Journal of Specialized Translation, 6, 2006, 134-151.

Singh, N. and Pereira, A. (2005) The culturally customized Web site: customizing web sites for the global marketplace, Oxford: Elsevier.

Shreve, G. M. (2006) 'Corpus Enhancement and localisation', in Dunne, K. (ed.) Perspectives on Localisation, Amsterdam-Philadelphia: John Benjamins, 309-331.

Trosborg, A. (1997) 'Translating hybrid political texts', in Trosborg, A. (ed.) Analysing Professional Genres, Amsterdam: John Benjamins, 145-158.

Vaughan M. W. and Dillon, A. (2006) 'Why structure and genre matter for users of digital information: a longitudinal experiment with readers of a web-based newspaper', International Journal of Human Computer Studies, 64, 502-525.

Venuti, L. (1995) The Translator's Invisibility. A History of Translation, London/ New York: Routledge.

Wright, S. E. (2006) 'Language Industry Standards', in Dunne, K. (ed.) Perspectives on Localisation, Amsterdam-Philadelphia: John Benjamins, 241-278.

Yunker, J. (2003) Beyond Borders: Web Globalization Strategies. Indianapolis: New Riders.

Workflow Specification for Enterprise Localisation

Madeleine Lenker, Dimitra Anastasiou, Jim Buckley
Centre for Next Generation Localisation,
Localisation Research Centre,
CSIS Department,
University of Limerick,
Limerick

www.localisation.ie
Madeleine.Lenker@ul.ie, anastasiou@uni-bremen.de, Jim.Buckley@ul.ie

Abstract

Localisation concerns the translation of digital content and software, and their appropriate presentation to end users in different locales. Localisation is important because having software, a website or other content in several languages, and meeting several sets of cultural expectations is an important international marketing advantage. In the non-commercial sector, where information equality is deemed important, localisation also enables information access for less supported languages such as Galician. These requirements have led to increased localisation activities and act as a prompt to study how the localisation process can be optimised. A necessary pre-requisite for this, is a standardised localisation process. Such a process not only provides a basis for continuous quality improvement (CQI), but also has the potential to direct further research in the area.

In terms of representing and enabling this standard localisation process, workflow technology provides an interesting solution. Workflows are concerned with formalising task structure: the order of subtask invocation, synchronisation and flow of information through the tasks, the roles of people that perform the tasks and the tracking and reporting of tasks. Since the 1970s, workflow technology has expanded into fields like manufacturing and accounting and now is reaching the localisation industry.

A generic localisation process definition in workflow form would allow companies standard process guidance when faced with localisation tasks, and would provide researchers with a framework within which they can contextualise their research and identify new challenges. The potential automation offered by workflow engines provides the potential of real efficiency gains for companies.

This paper describes research towards the representation framework of enterprise localisation workflows. We document several important localisation-process issues identified in the relevant literature. We will also evaluate existing workflow languages as a means of representing such localisation processes at a later stage.

Keywords: *enterprise localisation, standards, workflow, workflow languages*

1. Introduction

Localisation has developed into a multi-billion dollar professional industry during the last decades. Companies have increasingly sought international markets to sell their products and localisation has proven to be a key to this endeavour (Esselink 2000). Generally speaking, localisation is the translation and adaptation of software, websites and all related digital content such as documentation for specific locales (Esselink 2000). A typical localisation project would include the full translation, linguistic and functional testing of the localised software application)of a system, its online help and its documentation, printed and/or digital. As such, translation is a core component of the localisation process. Related activities in localisation projects are terminology management, editing, proofreading, page layout, file conversion, and testing.

Nowadays, we have an increasing amount of web-based applications, and database-driven websites that need localisation into multiple locales and this has resulted in a considerable rise in the volume of

content to be localised. Hence, new approaches need to be found to cope with this increasing volume while maintaining and improving quality.

This paper will concern itself with the enterprise localisation context (section 2). More precisely, we focus on localisation issues (section 3) relevant to workflow management in enterprises; issues such as automation, linguistic resources, quality and recognition of the importance of personnel. In this context a workflow is defined as "a computerised facilitation or automation of a business process, in whole or part" (Hollingsworth 1995). Workflow management systems define, manage and execute these workflows using a software application (Hollingsworth 1995). Thus, defining this process as a workflow would enable potential automation and efficiency of the process, and this paper is a first step towards that definition.

In addition, the paper will examine the most suitable workflow languages for representing this localisation process as a workflow (section 4). It refers to a classification of workflow languages according to resources, data sets and functions, while section 5 is related to expectation of workflow management tools. In section 6, we summarise some challenges of the current processes and present a high level enterprise localisation process that will act as a framework for exploration of more detailed enterprise localisation workflows.

2. The Enterprise Context for Localisation

Localisation occurs in many contexts: profit and non-profit. For example, voluntary organisations are often interested in localising information content on websites to provide medical health information to third-world locales (e.g. Traducteurs sans Frontiers). However, this paper concerns itself with enterprise localisation, where localisation is performed by a commercial organisation, typically for market gain. Consequently, we define what we mean by an enterprise. An enterprise as largely defined by a sense of large scale or at least scalability (Sikes 2009, p.16). Texin (2002/12: 15) defines several attributes of companies who localise large volumes:

i) Strong belief in ultimate product success;
ii) Millions of units sold;
iii) Recall is significant cost;
iv) Recall can kill a product's market;
v) Large QA staff and thorough QA process;
vi) Large beta test, lots of value-added reseller (VAR) interest;
vii) Single source is imperative;
viii) Coding standards for developers in place, including internationalisation;
ix) Chief executive officer (CEO) knows international impacts bottom line.

3. Issues in Localisation

There are many important issues in localisation which need particular attention. In the following paragraphs we discuss 3 such issues: automation, linguistic resources, and recognition of localisers.

3.1 Automation

Today, efficiency and thus automation, is a core concern for localisation. The so-called Translation Environment Tools (TEnTs) automate the translation process and Zetzsche (2009, p.200) presents the features which every TEnT tool should have:

i) resource (such as translation memory) lookup;
ii) terminology management;
iii) project management;
iv) word counts;
v) software localisation.

Sophisticated TEnT tools like memoQ from Kilgray also offer a complete translation workflow online via features such as online document storage and the ability to work online on a project together with other translators and people within the translation workflow (for example reviewers and terminologists). This kind of technology allows several people to work on the same project simultaneously. Data is also stored online and cannot get lost in a local computer crash. However, it should be noted that while these tools to create workflows exist, standard, state-of-the-art workflows do not seem to exist and consequently, companies are left to their own devices to recreate best practices in this regard.

3.2 Linguistic Resources

Linguistic resources are generally important to accelerate and facilitate the localisation process. In this subsection we refer to three different linguistic resources: translation memories (TMs), machine translation (MT) systems, and terminology databases.

With the adoption of web technologies in localisation, TMs can be accessed online from anywhere in the world and newly evolved exchange standards such as TMX allow data exchange between different computer-assisted translation (CAT) tools. Hence the advantages of TMs can be leveraged across companies, geographical locations and different platforms.

With the gain that TM implies, comes an associated need to maintain such linguistic resources. Dalmau (2009) states that TM quality always depends on the reliability of human translators. Errors that are not spotted will be recycled and made again and again. Therefore, the person responsible for the TM should perform maintenance to ensure high quality content as well as feedback to the end client (e.g. regarding terminology, style, etc). While this seems to be a lesser issue when done in-house, when localisation is being outsourced to localisation service providers (LSPs) and other contractors, it is more typical that the issue seems to snowball.

As for MT, its quality has improved over the years. Among many authors, Yunker (2008, pp.30-31) says that many companies now use statistical MT (SMT) in the first part of their translation workflows. SMT predicts how a sentence should be translated based on algorithms, and it learns from its "mistakes". SMT gets better when translations are fed to the system and the more closely aligned the engine is with the specific industry or application. Noteworthy here is that MT has high quality output for certain text types. For example, it is not suitable for legal or marketing texts, but it is suitable for text with high repetitions and controlled language use, such as technical documentation (Sonstenes 2009).

Another important linguistic resource is that of term bases and the associated activity of terminology management. Terminology management is the process of formalising the terms used within a company. According to the localisation company Lionbridge, terminology management aims to identify, capture, and deploy this unique terminology, and to do so localisers have to:

i) determine the scope of terminology management needs (product names, company internal terms);
ii) review any pre-existing glossaries, style guides, or formal communication formats to identify the core terminology base;
iii) establish a workflow process that catalogues, reviews, approves, and deploys new terminology to all stakeholders;
iv) establish an equivalent for each term within each target language;
v) provide access to the approved term base to internal authors, marketing communicators, external vendors and translators.

There were two surveys recently (Hurst 2009) carried out by SDL that underline the importance of terminology management throughout global content creation management. One survey was sent to translators to obtain their view on terminology and the other survey was sent to organisations such as Philips, Siemens, Cisco, etc. The key findings were that terminology is seen as very important but used inconsistently within several departments of organisations, which leads to an inconsistent source text. The surveys also show that organisations and translators do not manage terminology effectively, if at all. This is a weakness that needs to be explicitly addressed in enterprise localisation workflows. A terminology management process should be the basis for multiple localisation workflow instances since it provides the terminology to be used during several localisation projects. Therefore, it is a core component in localisation workflows.

3.3 Recognition of Personnel as a Key Issue

As localisation became a more recognised necessity for global enterprises, there was an increasing awareness of the importance of the people in localisation roles. Today a single localisation project typically involves several translators, editors, project managers, coordinators, engineers and people responsible for desktop publishing (DTP), to deal with various file types.

Rosner (2009, pp.26-27) surveyed practitioners to find out which was more important: people, technology or processes within localisation. The results indicated that the people and processes are the most important factors. Interestingly, none of the respondents thought that technology was the most important factor. To paraphrase, participants said if you have the right people and the right process in place, even bad technology will not stop you from achieving the goal. This is interesting given the investment in tools and technology in this area and the lack of standardisation with respect to a state-of-the-art localisation process/workflow that explicitly incorporates, and thus organises, people within that process.

Indeed, considering the cultural differences that can arise when the localisation effort is distributed globally (differences between headquarters and local offices and the different languages) this seemingly important factor could become even more critical, exacerbating the need for such a workflow. Steps such as those required to build trust and communication, in such contexts may be particularly desirable.

3.4 Quality and Standards

A core issue in many research fields, including software localisation is quality. Dalmau (2009) suggests that *we always get what we pay for* implying that, if we pay for a low-cost translation service, we cannot expect high quality. However, defining what we mean by 'high quality' is problematic. Dalmau reinforces this suggestion when she said that "we only tend to realize what quality is when it's missing" (Dalmau 2009, pp.51). However, according to O'Hagan (2004) there are two types of quality activities involved in the localisation process:

i) Quality Assurance (QA);
ii) Quality Control.

From one side, QA is the prevention of quality problems. QA can be heightened by planning and documenting activities. Therefore, quality management has to be in place to review and assess the system. From the other side, Quality Control's aim is to achieve and maintain the quality of a product, process or service. Essential for quality control is the monitoring of activities in order to find problems and to solve them.

These definitions imply that both QA and Quality Control of the entire process are essential for successful localisation, from the receipt until the delivery of a product (Doval 2005). However, quality can only be achieved and refined in the context of a known localisation process.

Rodríguez (2008, pp.44-46) writes about the implementation of quality in such a context and its accompanying quality management system (QMS). He finds that QMSs make a company more transparent: a company shows the quality process in place and ensures it as well. In terms of reliability, product defects tend to occur less when processes are in place to monitor the involvement of everybody in the company, setting clear goals and expectations.

In order to put a QMS in place, several principles seem to be used, such as step-by-step planning, company-wide involvement and an assessment of existing processes. Rodriguez points out that it is common that the QA of a company is based solely on its production process, neglecting sales, training, human resources etc. Employees were asked, for instance, how orders are created, how jobs are cycled through the system, how a project is invoiced but there was no mention of the non-process elements in such questions. He asserts that a clear definition of who is doing what, why and how, must be highlighted. This suggests that the workflow should not only discuss the procedures in place but should also define the roles of the personnel involved in these procedures, as suggested in section 3.3.

3.4.1 The Growth of Certifications/Standards

In industry, certification has become an important QA 'tool'. McBride (2002, p.10) points out that "companies who certify their applications enjoy greater market awareness and that they also benefit from 24x7 working efficiency - the most important enterprise benefit of all." In his view, a commitment to quality includes not only technical requirements, but also the public availability of the results (i.e.: certification). This can lead to a quality product association.

Given the importance of client confidence in a vendor's localisation quality, certifications for quality management and process evaluation have been made under the ISO 9000 standards umbrella (ISO 9001, 9002, 9003) and DIN 15038 (see Doval 2005). Certification is based on adherence to these defined standards and thus it helps to set up certain quality thresholds in how to perform a process, the associated products, services and people.

It should be noted, however, that standards in the localisation world (see the LISA QA Model for translation) only define the QA and the localisation process in general. They do not necessarily guarantee a good translation outcome. A process, for instance, cannot replace the project manager who picks the translator for a job; it only defines the procedure of selecting a translator. Therefore, a quality check of a translation cannot be treated like a check of a manufactured product.

4.Workflow Languages

Given the agenda of creating an enterprise localisation workflow, the question then arises as to the best way to present these elements in a workflow.

That is, what would be an appropriate workflow language and workflow management system?

The following paragraphs deal with two workflow description languages: Business Process Execution Language (BPEL) and Yet Another Workflow Language (YAWL). Both, BPEL and YAWL, are workflow description languages, expressed in Extensible Markup Language (XML). This fact, combined with tool support, means that they are interchangeable. BPEL is the industry standard and YAWL is picked as a growing-in-popularity alternative approach (AL Rossais 2005, p.22).

4.1 Business Process Execution Language (BPEL)
BPEL is a workflow description language to specify and execute workflows (van der Aalst et al. 2005). It "is supported by a lot of enterprise resource planning (ERP) and Workflow Management System Market leaders" (AL-Rossais 2005). According to Recker (2006, pp.521-532), the core concepts of BPEL are the following:

- Variables to store process data and exchange messages with web services;
- PartnerLinkTypes to define the required ports for message exchange;
- Basic Activities to specify operations to be performed in a process (invoked, received, replied web service operations);
- Structured Activities for the definition of control flow, e.g. to express synchronisation;
- Handlers in the case of faults

Among others, Brogi (2006) believes that BPEL is the most widespread language for composing web services which, according to the World Wide Web Consortium (W3C) are software systems designed to support interoperable machine-to-machine interaction over a network (typically Application Programming Interfaces (APIs)). But BPEL lacks formal semantics. It emerged from previous specifications such as Business Process Modeling Notation (BPMN), Web Services Flow Language (WSFL) and others (Glatard et al. 2009, p.13). Like WSFL, for instance, it includes control as well as data links. BPEL furthermore, has a locator element; this allows a service to be static, local or mobile.

4.2 BPEL4People
BPEL4People is an add-on for BPEL to support/execute and standardise human tasks with BPEL (Zhao 2008). BPEL4People was suggested by IBM, SAP and other companies to provide People Activities (human tasks) within existing BPEL processes. This add-on also addresses some recently occurring problems with BPEL, such as task authorisation.

So this variant of BPEL was extended with human tasks which can be invoked as web services (as BPEL uses individual web services as atomic elements (Zhao 2008, p.195). The human task is associated with a person by specifying roles and defining the permissions associated with the task, using a potentialOwner element (OASIS 2010). In this case, a person must notify the business process engine when a task is finished, either successfully or unsuccessfully. In order to execute and monitor processes, a task list is used. Tasks can have a status of hold, queried, claimed, revoked and failed (Zhao 2008, p.196).

BPEL4People also introduces the concept of PeopleLinks to bind a group of people to a business process (similar to PartnerLinks - see 5.1). This concept allows the suspension of a business process until a task, assigned to a person or a group, is completed. Interestingly, BPEL4People also allows constraints (authorisation constraints) to exclude users from performing certain tasks due to bad performance in previous tasks.

4.3 Yet Another Workflow Language (YAWL)
This workflow language was developed as a joint effort by the university of Eindhoven and Queensland University of Technology. It is an execution language with the goals of handling complex data transformations and web service integration. Brogi (2006) states it is the 'lingua franca' with which to express web services.

YAWL is based on high-level Petri Nets (advanced synchronisation, multiple instances and cancellation patterns are added, therefore defining extended workflow nets, Glatard et al, 2009, p.14). Petri Nets are designed "for modelling, analysis and simulation of dynamic systems with concurrent and non-deterministic procedures" (List and Koherr; 2006, p.1535) and so they provide these additional functionalities. They provide these on the basis of formal semantics and thus offer an abundance of analysis techniques (van der Aalst et al. 2005, p.247; Brogi 2006).

YAWL's workflow specification is based on a set of extended workflow nets which are in a hierarchical order. There are two kinds of tasks: atomic and

composite tasks (van der Aalst 2005, p. 253). Each composite task refers to a set of workflow nets that are at a lower level in the hierarchy. There is also a top level workflow in the extended workflow nets called the root of the tree-structure (van der Aalst et al. 2004).

4.4 Classification of Workflow Languages

Glatard et al. (2009) report a classification for workflow languages (see Table 1 below). They believe that if a workflow definition gathers functions, data and resources, then it is fully executable. If only functions are defined, but no data or resources are available in the description, the workflow representation is a functional one. If only data is defined and resources are not, we talk about a service workflow representation and if functions and data are defined without resources, we talk about a task graph (Glatard et al. 2009, p.5).

Defined	Formal models	Functional WF	Service WF	Task graphs
Resources	Not defined	Yes	Yes	No
Data sets	Not defined	No	No	Yes
Functions	Not defined	No	Yes	Yes

Table 1: Classification of workflow languages

BPEL/BPEL4People is a service workflow where functions and resources are specified. YAWL is considered to be a service workflow: Even though YAWL is built on the Petri Nets formalism, its goal is "to overcome the expressiveness limitations of the contemporary workflow management systems" (Glatard et al. 2009, p.14) and it acts like service workflows (van der Aalst & ter Hofstede 2002).

Ko (2009), providing a summary of all prominent workflow languages and their status, shows BPEL as an industry standard and an execution language as well as a stable standard. In contrast, YAWL has an academic background, but also is an execution language. It is not yet a standard, but is considered stable.

5. Workflow Tool Expectations

Alonso (1996) declared that "workflow management systems would become the technology of choice to implement large and heterogeneous distributed execution environments where sets of interrelated tasks can be carried out in an efficient and closely supervised fashion".

Furthermore, Alonso (1996) pointed to the appealing characteristics of this technology: decentralisation of the corporation, decentralisation of the decision making, satisfying the need for very detailed information about daily activities, the increasing availability of distributed processing technology.

Since the demand for workflow systems is increasing, more tools have been developed but the tools are now facing the demands of scalability and system-wide reliability. In the following paragraphs, we will deal with three characteristics that seem to be the major issues for such tools/systems (Muth et al. 1998): scalability, synchronisation and fault handling.

5.1 Scalability/Performance

Since we deal with enterprise localisation, we are concerned with huge amounts of translatable content, and the involvement of many people. Therefore, the system should be able to cope with many tasks (often in parallel) which are mainly automated (Alonso 1996, p.5). According to Alonso, the system must have architecture appropriate for this. He also implies the necessity for component design in a modular fashion for further system customisation (Heinis et al. 2005). Muth et al. (1998) state that scalability dictates the overall workflow processes and must be distributed across multiple workflow engines running on different servers. This partitioning may also lead to a partitioning of workflows (Muth et al. 1998, p.2).

5.2 Synchronisation/Parallelism

To express parallelism in a workflow is of high interest for grid-enabled workflows (Fox and Gannon 2006). In the current Grid context, they define a workflow as follows: "The automation of the processes, which involves the orchestration of a set of Grid services, agents and actors that must be combined together to solve a problem or to define a new service". For non-task graph workflows, according to Glatard et al. (2009), BPEL was recently enriched with a foreach operator to include parallel control structures. It can further distinguish between

parallel constructs where the number of tasks is known beforehand and where the number of tasks is unknown beforehand (and can only be discovered during execution). For this, an alternative was developed where data parallelism is implicit in the language. This means that only data flows are represented and the data description is externalised. The process description and the data sets to be processed lead to a parallel data flow (Glatard et al. 2009, p.20). The ability to synchronise concurrent data flows is important, as every workflow language handles data synchronisation differently.

5.3 Fault Handling

In workflow environments a system needs to handle exceptions and faults such as when an operation is cancelled unexpectedly. Errors can be unpredictable and can lead to a system crash. Therefore, a notification mechanism is needed. BPEL for instance supports different types of fault handling which provides a good recovery mechanism. According to Akram et al. (2006), the 'fault' can be network problems or unavailability of services. Another characteristic of 'fault' termination is clean up. Akram et al. (2006) point out that BPEL supports such clean-ups, through so called compensation handlers in an event (fault).

6. Prototype Workflow Challenges

> *"The key issue is the business process, and how that process works for the business users, partners, suppliers, and customers. Workflow languages are technologies that manage the definition, implementation, and operation of processes. One is not good and the other bad, but they come from a different origin, and thus have different strengths. The key is to look beyond the product name, and find the functions that will best serve the business."*
> Workflow Handbook 2005 (2005:22)

In the following sections we will look at the challenges of generating a literature review to find localisation elements that can be be used in conjunction with workflow languages, how this can be represented and we will also present a preliminary framework.

6.1 Literature Review

Generating a literature review for this subject was difficult in that much of the available material on workflows is contained within articles on other subjects, and as such is not prioritised.

In terms of our methodology, we reviewed localisation publications from 'MultiLingual' and 'Localisation Focus - The International Journal of Localisation' as well as several other books and journal articles from industry and academic experts, to see the literature status of workflows' existence in enterprise localisation. Only a limited number of articles were found, over a review period of ten years, which were relevant to our agenda. (We rejected papers reporting on tool presentations, irrelevant surveys, and company specific articles.). These were articles that reported on elements of localisation processes relevant to enterprise localisation. The identified 'Enterprise Localisation elements' were augmented by additional data from practitioners who identified several of their workflows to us. This is the basis on which our initial workflow abstraction, presented below, was derived. But it should be noted that further work should probe this prototype model in greater detail.

6.2 Workflow Representation

Something that all workflow languages have in common is that they differ significantly in their concepts and semantics which is due to the lack of an agreement on a formal basis for workflow languages. The WfMC (Workflow Management Coalition) made an approach but failed to establish a standard (v.d. Aalst, 2004).

There are two approaches to cope with these workflow language challenges: to either develop an extension (e.g. BPEL4PEOPLE) or to combine languages/systems in order to re-use existing resources (Bernauer et al. 2003). At this stage of the research, a combination of approaches, based on BPEL, would seem to be the best solution (Fischer 2005, pp.20-21). The Workflow Reference Model of the Workflow Management Coalition (1995) declares that it is common to combine a set of applications in order to provide the best and most efficient output. We have also seen that Enterprise Resource Planning (ERP) tools (Wortmann 2006) are used to execute appropriate project management for enterprises (e.g. accounting, invoicing, order tracking, etc.); therefore, a combination of an ERP tool with a workflow language is desired for completeness. BPEL is a commonly used industry standard and seems to facilitate easier integration of ERP tools; this is the reason we support BPEL primarily as the basis for our combination of languages. However, further research needs to be done to substantiate this claim and to see if other workflow languages are more suited to the extended workflow defined.

6.3 Preliminary Framework

We now propose an initial prototype framework for Enterprise Localisation within which enterprise localisation workflows can be defined (See figure 1). The first step is to define the process (for instance what to localise, resources and budget). This leads to the project enactment/kick off. This, in turn, leads to a decision on how to proceed: either using an ERP tool such as Plunet BusinessManager (for accounting, invoicing, project management) or using a translation management system (TMS) such as GlobalSight or SDL WorldServer. When the ERP tool is used, it will provide a more holistic solution giving, for example, better monitoring of the depreciation and appreciation of company assets. If the ERP solution is chosen then a connection should be made to TEnTs to process the localisation tasks further. The defined process ends with the delivery of the localised content and the invoicing of the service provider. The overall process is monitored in case of faults, delays or cancellations. In our framework, another element entitled 'Personnel monitoring and feedback' is added. Personnel are a key issue as seen in section 3.3 and their involvement should be present throughout the whole workflow. Feedback refers to both communicating the feedback and subsequent training.

Workflows for tasks such as terminology management, layout or file engineering are not shown in Figure 1 and will be the subject of further studies. Since the figure shows a high level process, the connection of e.g. the ERP tool and a translation tool is not made explicit.

Likewise, as linguistic resource management lies beneath both the ERP/TEnT tools and TMS nodes/processes, it is not included at this level of granularity, but should appear in subsequent refinements of the workflow elements.

In our opinion, it is unlikely that existing workflow languages are sufficiently rich to represent the localisation process required, but further research is needed in order to show, in detail, the struggle in handling all of the criteria specified above to a sufficient extent for enterprise localisation. One issue may be the interoperability of different localisation components e.g. to execute project management. This topic, along with the lack of standards and the establishment of standard procedures, requires further research.

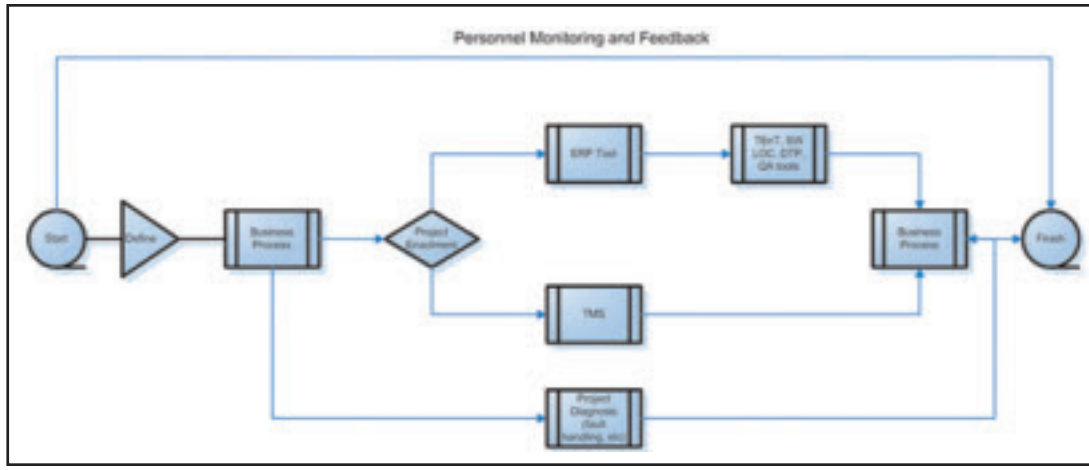


Figure 1: Workflow Prototype at a high level

In investigating Translation Memory Systems (TMS), we discovered the missing project management features that a best of breed ERP tool should have. TMS handle certain types of automation without manual interruption but speed and flexibility need further enhancement. These systems are often web based and may have a Java editor to set up workflows within the system. This is a good approach to access the TMS and to implement workflow options but those systems lack performance when it comes down to the actual translation process within the system itself. It can take up to several seconds to move on to the next translation unit. In the final analysis, this may lead to less productivity and lesser translations.

7. Summary and Conclusion

This paper demonstrates the importance of defining a standard workflow for enterprise localisation. We started with an introduction to localisation and focused on enterprise localisation. After referring to the characteristics of an enterprise, we then described some localisation issues relevant to and important for the enterprise localisation process, i.e. automation, linguistic resources, and recognition of localisers. Later the paper covered some standards of QA and their growth and adoption over the years.

A large part of the paper pertained to workflow languages, more precisely BPEL and YAWL. Also workflow tool expectations and specifically scalability, synchronisation and fault handling were examined. We concluded with workflow language challenges and the solutions to overcome them; our workflow suggestion covers the current business process and includes our recommendations for better monitoring.

References

Akram, A., Meredith, D., and Allan, R. (2006) 'Evaluation of BPEL to Scientific Workflows', Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, 269-274.

AL-Rossais, N. (2005) Workflow Management, Literature Review, King Saud University, College of Computer and Information Science, Computer Science Department.

Alonso, G. and Scheck, H.J. (1996) 'Research Issues in Large Workflow Management Systems', Proceedings of NSF Workshop on Workflow and Process Automation in Information Science, 126-132.

Bernauer, M., Kramler, G., Kappel, G. and Retschitzegger, W. (2003) 'Specification of Interorganisational Workflows - A comparison of approaches', Proceedings of the 7th World Multiconference on Systemics, Cybernetics, and Informatics, 30-36.

Brogi, A. (2006) From BPEL Processes to YAWL Workflows, Springer-Verlag.

Dalmau, M. (2009) 'The Importance of TM maintenance', MultiLingual, 49-51.

Doval, J. J. A. (2005) 'The EN-15038 European Quality Standard for Translation Services: What's Behind It?', Globalization Insider, available: http://www.lisa.org/globalizationinsider/2005/04/the_en15038_eur.html [accessed 27 June 2010].

Esselink, B. (2000) A Practical Guide to Localisation, Amsterdam & Philadelphia: John. Benjamins.

Fischer, L. (2005) Workflow Handbook, Future Strategies Inc.

Fox, G. and Gannon, D. (2006) 'Workflow in Grid Systems', Concurrency and Computation: Practice & Experience, vol. 18, 1009-1019.

Glatard, T., Montagnat, J., Bolze, R., Desprez, F. (2009) On scientific workflow representation languages, Report of Laboratoire d'Informatique de Signaux et Systemes de Sophia Antipolis, available: http://www.i3s.unice.fr/~mh/RR/2009/RR-09.10-J.MONTAGNAT.pdf [accessed 27 June 2010].

GlobalSight, Translation Memory System (Welocalise), available:http://www.globalsight.com/ [accessed 16 April 2010].

Heinis, T., Pautasso, C. and Alonso, G. (2005) 'Design and Evaluation of an Autonomic Workflow Engine', Proceedings of the 2nd International Conference on Autonomic Computing, 27-38.

Hollingsworth, A. (1995) Workflow Management Coalition, The Workflow Reference Model, Document Number TC00-1003, Issue 1.1.

Hurst, S. (2009) 'Wake up to terminology management', Communicator, 14-16.

IBM, The Web services insider, Part 4: Introducing the Web Service Flow Language, James Snell, Software Engineer, IBM, available:http://www.ibm.com/developerworks/library/ws-ref4/ [accessed 16 April 2010].

ISO, International Organisation for Standardization, Management standards, available:http://www.iso.org/iso/iso_catalogue/management_standards.htm [accessed 16 April 2010].

Kilgray Translations Technologies, memoQ, available: http://en.kilgray.com/ [accessed 16 April 2010].

Ko, R., Lee, S., Wah Lee, E. (2009) 'Business process management (BPM) standards: A survey', Business Process Management Journal, Vol. 15 No. 5, 744-791, available:ryanko.files.wordpress.com/2008/12/bpmj_stds_survey_koleelee.pdf [accessed 27 June 2010].

L10NBridge, Terminology Management, Standardize Your Business Terminology, available:http://www.lionbridge.com/lionbridge/en-US/services/localisation-translation/terminology-management.htm [accessed 03 May 2009].

LISA QA Model, available: http://www.lisa.org/LISA-QA-Model-3-1.124.0.html [accessed 27 June 2010].

List, B. and Korherr, B. (2006) 'An Evaluation of conceptual Business Process Modelling Languages', Symposium on Applied Computing, Proceedings of the 2006 ACM symposium on Applied computing, 1532-1539.

McBride, P. (2002) 'The Value of Certification for Multiple Locales', Localisation Focus, 9-10.

Medicines sans Frontiers, NJO, available: http://www.msf.org/ [accessed 10 April 2010], see also http://tsf.eurotexte.fr/?lang=fr [accessed 22 October 2010]

Muth, P. et al. (1998) 'From centralized workflow specification to distributed workflow execution', Journal of Intelligent Information Systems, 10(2), 159-184.

OASIS, Advancing open standards for the information society, available:http://www.oasis-open.org/home/index.php [accessed 16 April 2010].

O'Hagan, M., (2004) 'Putting the QA stamp on translation', LISA, Best Practice Guide, 29ff.

Plunet GmbH, BusinessManager (ERP tool), available: http://www.plunet.de/ [accessed 16 April 2010].

Plone, Documentation, Add-on Products Developer Manual, available:http://plone.org/documentation/manual/add-on-products-developer-manual [accessed 23 October 2009].

Recker, J. (2006) 'On the translation between BPMN and BPEL: Conceptual Mismatch between Process Modelling Languages', Proceedings of the 18th International Conference on Advanced Information Systems Engineering, 521-532.

Rodríguez, A. J. M., Almeida, A. and Andrada, R. (2008) 'Quality management systems in tourism: the spanish case', International Journal of Management Cases, 10(3), 503-511.

Rosner, G. (2009) 'What's trust got to do with translation?', MultiLingual, 26-27.

Schäler, R. (2007) 'Reverse Localisation', Localisation Focus, Vol. 6 Issue 1.

Sikes, R. (2009) Across Language Server v5, MultiLingual, 16.

Sonstenes, L. (2009) 'Näher am Anwender, Maschinelle Übersetzung im Lokalisierungsprozess', Technische Kommunikation, tekom, 31. Jahrgang, Heft 02/09, 26-31.

Szirbik, N., Wortman, H. (2005) 'ERP and Workflow Systems: Do they work together?', Proceedings of IFIP 5.7 Conference on Advances in Production Management Systems (APMS 2005).

Texin, T. (2002) 'Who speaks for the little guys?', Localisation Focus.

Van der Aalst, W.M.P., Ter Hofstede, A.H.M., 2005. 'YAWL: yet another workflow language'. Information Systems, 30(4), 245-275.

Van der Aalst, W.M.P., Dumas, M., Ter Hofstede, A.H.M. Russell, N., Verbeek, H.M.W. and Wohed, P. (2005) 'Life after BPEL?', International Workshop on Web Services and Formal Methods (WS-FM), 35-50.

Van der Aalst, W.M.P., Ter Hofstede, A.H.M, Aldred, L., Dumas, M., 2004. 'Design and implementation of the YAWL system', Lecture Notes in Computer Science, Eindhoven University, 281-305.

Van der Aalst, W.M.P., Ter Hofstede, A.H.M, 2002. 'Workflow patterns: On the expressive power of (Petri-net-based) Workflow Languages', Proceedings of the Fourth International Workshop on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark, August 28-30, 1-20.

W3C, Web Services http://www.w3.org/2002/ws/ [accessed 30. September 2010]

Yunker, J., (2008), 'The end of translation as we know it', MultiLingual, 30-31.

Zetzsche, J. (2009), ToolkitV7.

Zhao, X., Qiu, Z., Cai, C., and Yang, Y. (2008) 'A formal Model of Human Workflow, Web Services', ICWS '08. IEEE International Conference, 195-202.

Automated text generation for the localisation of an online game

K.A. Arthur, B. Brandt, A. Fedane, D. Hannan
Brandt Translations,
3 Jocelyn Mall,
Jocelyn Street,
Dundalk, Co. Louth.

info@BrandtTranslations.com
www.BrandtTranslations.com

Abstract

The problem being examined arises in the gaming industry, where the addition of new and varied content is of great importance to gamers. New content addition is a cost driver and so, finding ways to allow scalable content, while managing costs will be of interest to producers. A solution is presented in the paper to the problem of localising a large volume of sentences of text with similar structures. An automated text generation tool was written in Ruby on Rails, which implemented a set of grammatical rules for each of the project languages, and generated grammatically correct text to appear in the user interface of the online game. The costs of full product internationalisation and rules based translation are compared in a simple model. We conclude that the rules based approach to automatic text generation is less costly, and scalable for this particular case.

Keywords: *String concatenation, localisation translation, cost reduction, translation review, string parsing, grammar rules, online games.*

1. Introduction

Automated telephone attendants, that provide information, such as, "You have " + " 2 " + " messages" are very common. They produce sentences by concatenating parts of sentences together to form whole and grammatically correct sentences. The English language is relatively easy to work with when it comes to using rules for simple sentence formation. In languages other than English, the rules for forming simple sentences are more complex and context dependent.

The purpose of this paper is to describe the solution we implemented to manage the translation cost of the localisation of a massively multiplayer online text based role playing game (MMORPG). The problem we faced was the translation of large sets of sentences that were structured very similarly, contained similar content and that needed to be scalable. In this context, the scalability of the translation content referred to the ability to add more content as desired. The software on which we were working was not internationalised, and the generation of much of the content visible to the user was through string concatenation, whereby strings are added together to form sentences. While this is appealing in English, and there are very few grammatical exceptions for which to cater, when we move to other languages, the localisation of such content becomes problematic.

String concatenation is to be avoided, according to internationalisation best practice (see Brandt 2010, Akkas 2010). In general, there should be one string to localise in the resources for every instance of a string appearing in the software user interface. If we were to implement best practice for this project, we would have had to rewrite significant portions of the base code, translate and review the software and then perform functional testing. Each phase would incur significant costs.

The problem we faced was to generate a solution that minimized the costs of producing translations for a

large volume of words, without incurring both large translation cost and large review cost. Our solution involved analysing the text of the source sentences per language, and identifying a general structure for each language to which each sentence adhered. The individual words were translated and a rules based Ruby parser was used to output the target sentences.

We put the target languages sentences in spreadsheets so that translators could easily review the text. This avoided having to bring up each sentence exhaustively in gameplay. Some gameplay was necessary so that the translators could see a selection of the sentences in context.

2. Problem description

In this section, we describe the problem in greater detail and give specific examples of the text to be translated. We also examine the grammatical construction of the sentences and show their structure. The example languages we considered were; English, Spanish, Russian and German.

2.1 String concatenation

String concatenation is the process of adding together strings to make a whole message. The process is used in many applications, for example, in telephone automated attendant software, we have:

You have + one + new message.
You have + two + new messages.
You have + six + new messages.

In the online game, some of the example text is as follows:

Hero+defeated+a Cyclops+in+The Abandoned Caves of Mumbo
Lady+was defeated by a+Cyclops+in+The Abandoned Caves of Mumbo
You are on a quest in the+The Baneful Dungeons of Bella+searching for+The Scroll of Bella

The English examples shown above demonstrate the versatility of the English language and that adding together components can make regular grammatically correct sentences. The only irregular sentence in the telephone attendant example is the first sentence. The second and subsequent sentences match the number with the plural of the word message, simply by adding the letter "s". There is no such simple solution in other languages.

2.2 Case

Part of the reason for the complexity of the translation from one language to another is the fact that nouns may change depending on how they are used in the sentences. Case is the grammatical name for this process. B.J Blake (2001), states that, "Case is a system of marking dependent nouns for the type of relationship they bear to their heads". The author goes on to say that case "marks the relationship of the noun to a verb at the clause level or of a noun to a preposition…" For our example, this means that nouns change their form (decline) depending on how they appear in the text. Case is not the only grammatical process to cause a noun to change its form. In addition, a noun can change its form to agree with the verb to match; case, number or gender. In English, case is not a significant feature of the language, compared to Latin, Greek or other Indo-European languages. Although the purpose of this article is not to discuss grammar in-depth, enough will be described of the main concepts in order to implement our semi-automated solution for MMORPG localisation.

The context of our problem is the localisation of a massively multiplayer online, text based fantasy adventure game, into German and Spanish. An example of the text that needed to be translated is as follows:

You have found+an+ancient+Kovalli+copper+braclet
You have found+an+ancient+Kovalli+gold+braclet
You have found+an+ancient+Kovalli+gold+chalice
You have found+a+new+Mithican+bone+chalice
You have found+a+new+Mithican+bone+cup

There are other types of text in the game, providing similar issues, but for the purposes of this paper, we will examine only the above in detail. For ease, we will refer to the sentence as the "loot sentence".

Examples of case categories in language are as follows: nominative, accusative, genitive, dative, locative, and instrumental cases. The following list shows examples of how these cases are used in sentences, (BBC 2001 online);

Case	Example
Nominative indicates the subject	"*John* buys the newspaper."
Accusative indicates the direct object:	"John buys the *newspaper*."
Genitive indicates possession:	"*John's* newspaper is cheap."
Dative indicates the indirect object	"John writes *to the newspaper*."
Locative indicates location or topic of conversation and is governed by the prepositions 'in', 'on', 'about', and 'in the presence of:	"John talks *about the newspaper*."
Instrumental has the sense of 'with' or 'by means of':	"John swats a fly *with the newspaper*."

Case	English	Russian
Nominative Case	Ivan sees the table.	Иван видит стол.
Accusative Case	Ivan closes the book.	Иван закрывает книгу.
Genitive Case	Ivan wrote a letter to a friend of Boris with a pen.	Иван написал письмо другу Бориса ручкой.
Dative Case	Ivan gave the table to Valery.	Иван дал стол Валерию.
Prepositional Case	The book is on the table	Книга—на столе
Instrumental Case	Ivan wrote a letter to (his) friend with a pen.	Иван написал письмо другу ручкой.

Table 1 showing examples of case in English and Russian (Lexiteria 1996 online).

As mentioned above, case, number and gender cause nouns to change their form. Each of the languages we consider in this paper is listed in Table 2 along with the number of cases and number of genders of nouns.

Language	Number of genders (Masc/Fem/Neut)	Number of cases
Spanish	3	0
German	3	4
Russian	3	6

Table 2 showing the list of languages, the number of noun genders and the number of cases in the language.

Just to note, Spanish has 3 genders of noun, but the neuter gender is not commonly used.

2.3 Software

When logged into the online software, the user interacts with characters in the game, picking up points, loot and other items, while visiting a number of places, or "locations", in the fantasy world. In the next sections, we will describe the "loot" and examine the treatment of the "loot" sentence in each language in turn. We identify a general structure for the loot sentence in each language and show how the sentences are generated. We implemented similar structures for the location names and the creatures inhabiting the game, but for simplicity and to avoid repetition we will describe only the loot. Later in the paper, we describe the implementation of the "rules engine" for generating sentences.

2.4 General sentence structure

An adventurer may win or find some loot and this information is presented in the game user interface. Examples of the types of sentence include; "you found an ancient bone Accardian bracelet" or " you found a new bronze Kovalli cup". There are structural variations within sentences, and some sentences have common words. Examples of English sentence templates are as follows:

Example structure:
{age}{material-jewellery}{source}{loot_name}

Example sentence:
ancient + bone + Accardian + bracelet

Example structure:
{size}{type} of {kind}

Example sentence:
huge + chest + of + cash

Example structure:
{material-quality} {source} {loot_name}

Example sentence:
magical + Gurthian + cup

The template structure can be reordered, as the word order may vary from one language to another. For each sentence template element, we must maintain a table of its masculine/feminine/neuter, singular/plural and case forms. The "rules engine" uses the given sentence structure for each language and generates the appropriate sentence. All of the sentence elements are stored in categorised tables, which are looked up as required.

In the following sections we will examine examples of sentence construction in the different languages. We will also demonstrate via a short calculation that even with a modest number of variables, the number of sentences to translate can quickly grow.

2.5 Spanish

The Spanish language has two genders for nouns, namely masculine and feminine, and no cases. The sentence structure in English for the sentence to describe treasure that you found is, for example; " ancient bronze Accardian brooch":

{age} {material-jewellery} {source} {loot_name}

In Spanish the order of the sentence has to be rearranged, to the following;

{age} {loot_name} {material-jewellery} {source}

The {age} variable declines with the gender and number of the variable {loot name}. This means that the words signifying age, such as "ancient" or "new" will vary their form depending on the type of "loot" picked up by the adventurer. The {material-jewellery} variable does not vary. The {source} variable varies depending on the gender and the number of the {loot name}.

Attribute age en_US	M Singular	F Singular	M Plural	F Plural
Ancient	antiguo	antigua	antiguos	antiguas
Antique	histórico	histórica	históricos	históricas
New	nuevo	nueva	nuevos	nuevas

Table 3 shows examples of the "age" attribute of the loot.

Attribute material-jewellery en_US	Attribute material-jewellery es_ES
Bone	de hueso
Bronze	de bronce
Iron	de hierro
Silver	de plata
Tooth	de marfil

Table 4 shows examples of the "material-jewellery" attribute of the loot.

Loot name en_US	Loot name es_ES	Fem/Masc	Singular/Plural
Necklace	collar	M	S
Bracelet	brazalete	M	S
Earring	pendiente	M	S
Earrings	pendientes	M	P
Brooch	broche	M	S
Bangle	pulsera	F	S

Table 5 shows examples of the "name" attribute of the loot and grammatical gender (fem/masc) and number (sing/pl).

Attribute source en_US	M Singular	F Singular	M Plural	F Plural
Accardian	Acardiano	Acardiana	Acardianos	Acardianas
Gurthian	Gurtiano	Gurtiana	Gurtianos	Gurtianas
Kovalli	Kovaliano	Kovaliana	Kovalianos	Kovalianas

Table 6 shows examples of the loot "source" name and its variations to match gender and number.

Loot_type en_US	Loot_type es_ES	Gender	Number Sing/Plur
Hoard	pila	F	Singular
Chest	cofre	M	Singular
Pots	calderos	M	Plural

Table 7 shows examples of the loot "type" name and its variations to match gender and number.

Loot_size en_US	M Singular	F Singular	M Plural	F Plural
Colossal	colosal	colosal	colosales	colosales
Huge	enorme	enorme	enormes	enormes
Interesting	interesante	interesante	interesantes	interesantes
Impressive	impresionante	impresionante	impresionantes	impresionantes
Small	pequeño/	pequeña	pequeños	pequeñas

Table 8 shows examples of the loot "size" and its variations to match gender and number.

Loot_kind en_US	Loot_kind es_ES
Cash	monedas
Loot	botín
Treasure	tesoro
Valuables	riquezas

Table 9 shows examples of the loot "kind".

The game provides for a random element, which varies the type of treasure found by the adventurer. The "rules engine" takes the random element to identify the type of loot found, and associates it with an item in the loot table. Other random variables identify the source, age and material characteristics of the loot. The rules engine generates a sentence consistent with the grammatical rules of the language, specifying gender and number of the loot variable, as stored in the loot table. The age and material variables must hold content that is grammatically consistent with the loot variable content.

Examples of some of the Spanish sentences constructed from the template and data tables are as follows:

histórico brazalete de oro kovaliano
antigua pulsera de plata acardiana
nuevos pendientes de marfil gurtianos

In addition to the structure given above, there are also other "loot" sentence templates used in the generation of this type of text. This provides a rich environment in the game for the user.

The next type of loot sentence to be examined has the English template structure:

{size}{type} of {kind}

In Spanish the word order of the text remains as it is in the English. The variable {size} varies depending on the gender and the number of {type}. The variable {kind} does not change. Examples of additional loot sentence templates, with sample text are as follows in figure 10.

English template	Sample
{size} {type} of {kind}	Huge pots of gold
{loot_sentence} of {where_found}	Ancient silver Accardian bangle of Lars
{material-quality} {source} {loot name}	Magical Accardian cup

Table 10 shows examples additional loot template sentences in English.

The corresponding Spanish templates are ordered as follows:

Spanish template	Sample
{size} {type} de {kind}	enormes calderos de oro
{loot_sentence} de {where_found}	antigua pulsera de plata acardiana de Lars
{loot_name} {material-quality} {source}	copa mágica acardiana

Table 11 shows examples additional loot template sentences in Spanish.

In addition to the loot an adventurer might come across, the software also has a number of magical creatures, locations and other features, as mentioned earlier in this paper. The sentences for these features are generated analogously to the above.

Given a data set containing X number of variables for age, Y number of variables for item, Z number of material types and W variables for provenance, the sum total of combinations is XYZW. Consider the increase in the amount of content if one of the variables, for example X, is incremented by one. The increase in volume of sentences is:

$$(X + 1)\ Y\ Z\ W - X\ Y\ Z\ W = Y\ Z\ W \qquad (1)$$

When the number of variables of type age, that is X, is increased by one, then the increase in the number of sentences is: Y Z W. For even modest data sets, the final number of sentences that can be generated grows large quickly. For gamers, variety of content and continuous additions is of premium importance. New content when viewed in this way is a cost driver for the game. Finding ways of allowing scalable content, while managing to contain costs is of vital importance.

2.6 Russian

The Russian language declines nouns for gender, number and case. There are 6 cases in Russian, namely, nominative, accusative, genitive, dative, locative, and instrumental cases, see (SEELRC 2007). In addition, there are 3 genders for nouns; masculine, feminine and neuter. As for Spanish, we begin with the loot sentence structure for English, which is as follows;

English:
{age} {material-jewellery} {source} {loot_name}
The corresponding sentence template structure for Russian is as follows:
{age} {material-jewellery} {source} {loot_name}

In this example sentence, the structure and element order of the sentence template does not change from English to Russian, as it did for Spanish.

The loot appearing in this sentence is in the nominative case. All attributes for the noun describing the loot have the nominative form, where the ending of the attribute depends on the noun describing the loot name. For the purposes of illustration, we will display a limited amount of Russian data in the following tables.

Attribute age en_US	Attribute age ru_RU	Fem	Masc	Neuter	Plural
ancient	древний	древняя	древний	древнее	древние

Table 12 shows examples of the "age" attribute of the loot in the nominative case.

Attribute age en_US	Attribute age ru_RU	Fem	Masc	Neuter	Plural
ancient	древний	древней	древнего	древнего	древних

Table 13 shows examples of the "age" attribute of the loot in the genitive case.

Attribute material-jewellery en_US	Attribute material-jewellery ru_RU
bone	костяной
bronze	бронзовый
iron	железный
silver	серебряный
tooth	из зуба (no adjective in Russian)

Table 14 shows examples of the "material-jewellery" attribute of the loot.

Loot name	Loot name ru_RU nominative case	gender	number
necklace	ожерелье	Neuter	Singular
bracelet	браслет	Masculine	Singular
earring	серьга	Feminine	Singular
earrings	серьги	Feminine	Plural
brooch	брошь	Feminine	Singular

Table 15 shows examples of the "name" attribute of the loot and grammatical gender (fem/masc) and number (sing/pl).

Attribute source en_US	Attribute source ru_RU – nominative	fem	masc	neut	plural
Accardian	Аккардинский	Аккардинская	Аккардинский	Аккардинское	Аккардинские

Table 16 shows examples of the loot "source" name and its variations to match gender and number, nominative case.

Attribute source en_US	Attribute source ru_RU – genitive	fem	masc	neut	plural
Accardian	Аккардинский	Аккардинской	Аккардинского	Аккардинского	Аккардинских

Table 17 shows examples of the loot "source" name and its variations to match gender and number, genitive case.

In a fashion similar to Spanish, the rules engine for Russian generates loot, and other, sentences from data tables of translated words, ensuring that the sentences are grammatically correct. Grammatical correctness is ensured by matching the case, gender and number of constituent words in the loot sentences.

2.7 German

In general, the process for generation of the loot sentences in German follows the same structure as the Russian and Spanish languages. We will not repeat the detailed descriptions of text generation given for Spanish with German data, but due to German grammatical rules, we had to make additional changes to the template structure and rules engine.

Modern German has four declensions (nominative, genitive, dative, accusative) and three genders of nouns, masculine, feminine and neuter. Rather than discuss the exposition of the generation of loot sentences for German, we feel that at this stage, it is more valuable to discuss the exceptions arising in this language.

2.7.1 German Compound nouns

Given the English loot sentence template:

{age} {material-jewellery} {source} {loot_name}

The loot sentence template structure for German is as follows:

{age} {source} {material-jewellery} {loot_name}

Examples of loot sentences generated from this simple template are as follows:

Singular: neue akkardische Bronze-Halskette
Singular: altes kovallisches Gold-Armband
Plural: antike Frosthamer Silber-Ohrringe

Note, that the sentences chosen in this model are relatively simple, and do not contain grammatical structures containing pronouns, interrogative or negative clauses, as these were not required for the online content. In German, the template element order has to be changed to the following;

{adjective} + {source-material} + {noun}
new Frostham brass pen.

This means the material and the object will have to be grouped together for all such sentences. In German, these words form a compound noun. The formation of a compound noun has to follow certain rules. In most cases, the words for source and material become one word, such as "brasspen". However in the exception cases, the source and material words require an additional "s" to be placed where the two words join, which for our data set was not required. If this is required, the rules engine can be adapted for these special cases.

The loot sentences in German have to be generated in different cases depending on the context. In English we have the following sentence:

You have stumbled upon a battle between two groups of adventurers over {size}{type}of {kind}
You have stumbled upon a battle between two groups of adventurers over a Huge Hoard of Cash

The German translation requires an accusative object, so that the variables {loot_type} and {loot_kind} match. Examples of the variables; {loot_type} and {loot_kind} are as follows:

{loot_type}:
Cache, Chest, Hoard, Mound, Mountain, Pile, Stash, Stockpile,
{loot_kind}:
Cash, Loot, Treasure, Valuables, Zorkmids

When words such as, "Kiste" (case) or "Truhe" (chest) are paired with "Geld" (cash), "Beute" (loot) or "Schatz" (treasure), a compound noun is formed: {loot_kind}{loot_type}. For example: Geldkiste, Beutekiste, or Schatztruhe.

When words such as, "Kiste" (case) or "Truhe" (chest) are paired with, "Wertgegenstände" (valuables) or "Zorkmids", the following rule applies:
{loot_type} mit {loot_kind}
where {loot_kind} has to be in the dative case. This rule generates loot sentences such as;
Kiste mit Wertgegenständen
Truhe mit Zorkmids

If a noun denoting an unspecified amount is paired with "Geld" (cash), "Beute" (loot) or "Zorkmids", the following rule applies:
{loot_type} {loot_kind}
This rule generates loot sentences such as;
Menge Geld
Haufen Beute
Berg Zorkminds

If a noun denoting an unspecified amount is paired with "Wertgegenstände" (valuables) or "Schatz" (treasure), the following rule applies:
{loot_type} von {loot_kind}
where {loot_kind} has to be in the dative case. This rule generates loot sentences such as;
Masse von Wertgegegenständen
Berg von Wertgegegenständen
Unmenge von Schatzstücken
Haufen von Schatzstücken

German was not the only language requiring special rules to process certain types of sentence, but it required most of the special cases. The difficulties encountered in automated sentence generation shows explicitly that natural languages do not easily map to one-another in a mechanistic fashion, and that for every "general" grammatical rule there exist exceptions.

2.8 Rules engine

We have discussed the sentence template structure for the languages (Spanish, Russian and German) relevant to our project, and the process for generating grammatically correct sentences from components. For the limited "loot sentence" examples, we have also specified which sentence element drives the agreement with the other sentence elements for each sentence template type. We will now look at describing the rules engine algorithm showing example from the Russian language.

The content to be localised for the online game was analyzed by linguists, who identified the grammatical rules for each sentence type in the content. From our localisation experience, we know that languages do not map to another in a unique and one-to-one fashion, and for every attempt to define a mapping process there exists exceptions.

The sequence of events one would use to generate the loot text for the template: {age} {material-jewellery} {source} {loot_name}, is as follows:

1. From the game software random element, generate the loot name. This provides the gender and number to which other sentence elements have to agree. Example: ожерелье (necklace)
2. Generate adjective of type: age. Example: древний (ancient)
3. Select appropriate nominative adjective form from adjectives list. Example: древнее
4. Generate adjective of type: source. Example: Аккардианский (Accardian)
5. Select appropriate nominative adjective form from adjectives list. Example: Аккардианское
6. Generate adjective of type: material-jewellery. Example: серебряный (silver)
7. Select appropriate nominative adjective form from adjectives list – серебряное
8. Place the words generated in the sentence template.

Example output might be the following:
{age} {material-jewellery} {source} {loot_name}
– Ancient silver Accardian necklace
{age} {material-jewellery} {source} {loot_name}
– древнее серебряное Аккардианское ожерелье

The Rules Engine is a piece of software that uses the grammatical rules for the target language and puts together an appropriate string based on the grammatical rules for that string. It follows the above steps. The engine has an exception list of special cases, which are different for each language and sentence template type. Naturally, if the list of special cases was a similar number to the total number of sentences, then we would not consider the rules engine to be useful. The Rules Engine generates value because the number of special cases is significantly lower than the total number of cases that can be implemented with the generic grammatical rules.

2.9 Costs

To evaluate the costs of the project, we have to consider what alternative methods are available. In our analysis, there are two methods;

- Internationalisation - where every sentence containing a variation is stored as a single string, following the rule that there be no string concatenation.
- Rules engine approach - where single words are translated and concatenated into full strings.

For simplicity, and to facilitate comparison between the two models, we assume;

- The software internationalisation costs are going to be similar to the costs for the development of the Rules engine.
- The review costs of the translations generated by both the rules method and the full internationalisation method are similar.

This means that in this simple model, all we have to do is compare word counts from the two methods to get an overal indication of which is the lower cost. In the full internationalisation case, every variation in a variable generates a sentence.

Number of sentences = X Y Z W sentences (2)

Let there be N words per sentence on average, so that the total number of words is:

Number of words = X Y Z W . N words (3)

For simplicity we have discounted the effect of repetition in translation memory models. The effect of translation memory is to reduce the amount of new words. In this case then the average number of words per sentence would be reduced. This effect does not remove the multiplicative nature of the word count.

In the case of the rules engine, let the number of genders be g. In sentence templates where there are singular and plural instances, such as Table 6, a factor of 2 is picked up, since the number of words to be translated is doubled. The number of words for each variable in the sentence is;

age = 2 g X words. (4)

material = Z words. (5)

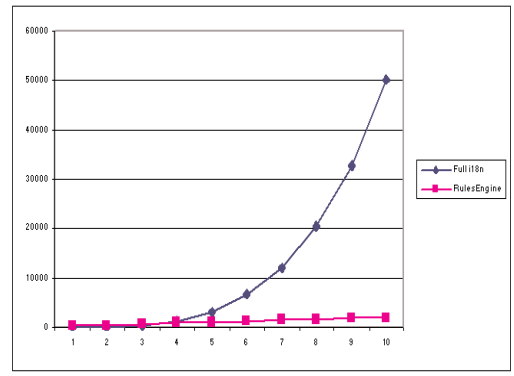
loot name = 2 Y words. (6)

source = 2 g W words. (7)

Total = 2 g X + Z + 2 Y + 2 g W words. (8)

To allow for multiple words in the variables, we will apply a factor of N, which is the average word count per sentence, to the total word count. The word count also picks up an additional factor when we add in the number of cases, but the total word count remains linear in the variables, X, Y, Z and W for the Rules Engine model.

Figure 1 showing the comparison of word counts for the different models.



As can be seen from Figure 1 the internationalisation model word count increases significantly faster than the Rules Engine approach which is linear in the variables. The internationalisation model, as described above, follows the Microsoft model (Microsoft 2010 online) where "Code doesn't concatenate strings to form sentences." The implication of this is that every string that is required for presentation in the user interface is stored as a whole. The calculations above compare the word count of increasing numbers of full sentences, versus the word count of increasing numbers of single words.

It is instructive to ask how valid are our assumptions? The first assumption compares software internationalisation costs and software development costs. In the case of software internationalisation and

software development the same team of resources could be used. Since cost is the rate multiplied by the time taken for the task, this means that the real comparison is between the amount of time required to perform the task of internationalisation for the product and the task of development of the Rules Engine. Both tasks would require a similar amount of development effort. Where this assumption fails, however, is in the implications of each task. Internationalising the product requires that the game is extensively functionally tested in all aspects. The testing for the addition of the Rules Engine only requires that a small part of the product be functionally tested.

The second assumption of the model is more robust to analysis. It states that the translation review costs are similar in both the software internationalisation and Rules Engine cases. In both cases the translators will review their translations both in text format, which occurs during translation, and in-context format, which occurs during game play. In the first case, the translators review their translations in something like MS Excel. For the in-context review, the translator reviews their translations in the context of game play. Howsoever the translations are generated and presented on the screen; the translator should not be able to tell how their strings are generated (if it is done correctly).

3. Conclusions

In this paper we have described the localisation problem we faced, and described the cost constraints prescribed by the client. We presented the approach taken to building a text generation application as part of an online game. The role of grammatical rules of case and gender matching of words, and the examples of special cases were described. An overview of the Rules Engine algorithm was presented. In the final section, we analysed the costs of the two available alternative approaches to the problem. Simplifying assumptions were made to facilitate comparison of the methods. In reality, a full internationalisation of the product code would require complete functional testing of the product from end to end, which was not required for the Rules Engine approach, as explained in the section above. The solution implemented above will, most likely, not work for all languages, but it does work for a sufficiently large set of languages and sentence templates for the purposes of our project. In summary, we were able to offer a cost effective method for allowing the addition of content in a scalable fashion in an online game, while managing to keep control of the key cost driver.

References

Blake, B. J., (2001) "Case", Cambridge University Press, http://www.amazon.com/Cambridge-Textbooks-Linguistics-Barry Blake/dp/0521014913#reader_ 0521014913 [accessed 22 Jun. 10]

Brandt (2010) http://www.brandttechnologies.com/media_dev_localization.shtml [accessed 23 Jun. 10]

Akkas, E., (2010) http://www.emreakkas.com/internationalization/10-internationalization-tips-for-developers-i18n-checklist [accessed 23 Jun. 10]

The Slavic and East European Language Research Center (2007) http://www.seelrc.org/ [accessed 22 Jun. 10]

BBC (2001) http://www.bbc.co.uk/dna/h2g2/alabaster/A622513 [accessed 22 Jun. 10]

Lexiteria (1996) http://www.alphadictionary.com/rusgrammar/case.html [accessed 13 Oct. 10]

Microsoft (2010) http://msdn.microsoft.com/en-us/library/cc194756.aspx [accessed 14 Oct. 10]

A Micro-Crowdsourcing implementation: the Babel Software Project

Chris Exton[1,2], Brendan Spillane[2] , Jim Buckley[1,2]
[1]Centre for Next Generation Localisation,
[2]Department of Computer Science and Information Systems,
University of Limerick

www.cngl.ie
www.localisation.ie
chris.exton@ul.ie; spillane.bren@gmail.com; jim.buckley@ul.ie;

Abstract

Today a lack of access to localisation resources and the cost of these resources contribute significantly to the digital divide. Therefore, finding ways to reduce cost and increase the rate at which software can be localised must be given high priority. While technologies such as Translation Memory and Machine Translation represent a major leap forward in improving this rate, they are, however, viewed mostly as enabling technologies and will not replace the human translator in most cases. This means that, the ability to acquire and manage an extremely large pool of human translators/post-editors is a major stumbling block in improving the rate at which software can be translated, as part of localisation. A Micro-Crowdsourcing architecture, proposed by Exton et al. (2009), could significantly reduce cost and increase capacity within the localisation space. For this reason the authors have created an exploratory implementation based on this architecture to investigate its feasibility.

This paper describes the exploratory implementation of the Micro-Crowdsourcing architecture called Babel Software. This contains two major parts, the development of the Babel Client Library and the Babel Server. The development process was demanding, yet found that the development of full Micro-Crowdsourcing should be feasible. It also found that, although it was possible to retro fit an existing application with the Babel architecture it is not advisable and so Babel should be incorporated into a client application's design during the initial development phase. An extremely important aspect of the implementation was based around the exploration of management tools on the Babel Server. These tools can be used by a project owner managing the localisation community. This helped provide great insight into the infrastructure which the Babel Server should provide in future implementations. Babel Software is an open source implementation which can be found on Souceforge.net. It is hoped that this exploratory implementation will spawn future releases; which will see Micro-Crowdsourcing complement other technologies in the localisation workflow. Together, these technologies can further reduce cost and increase the rate at which software is localised in an effort to reduce the digital divide.

Keywords: *localisation, digital divide, micro crowdsourcing, real time localization, micro-crowdsourcing, babel, open source, java, server*

Introduction

Most commentators agree that the concept of localisation refers to adapting a product to meet the language, cultural and other requirements of a specific target market or "locale" (Collins 2001; Ebben and Marshall 1999). Localisation in software systems involves more than simply translating the content exposed to users of such systems. It also involves adaptation to the local culture (time and date formatting, Number format, cultural conventions, etc…). However, language quality is fundamentally important, as content constitutes the core of these final products (N. Fernández 2000).

Today, there is great demand and value placed on localisation resources. Both commercial organisations and non profit organisations alike require these resources to access different locales. In the commercial sector, large US companies have found that income generated in foreign markets exceeds 65% of their total income (Asnes 2009).

Many of these companies produce or distribute software.

Non-profit or open source projects such as Open Office, also place great emphasis on trying to translate their software into as many languages as possible. There is great value to both types of organisations in being able to localise this software, albeit for very different reasons. Despite this, there is a significant gap between the locales that these organisations might wish to access and what is actually realised. This is highlighted by the fact that most digital information is presented in languages used by countries in the northern hemisphere (Schäler 2004). The lack of access to localisation resources and the cost of these resources is undoubtedly a significant contributor to this reality. This inability to access other locales has a negative effect on society as a whole and contributes significantly to the digital divide. Therefore, finding ways to reduce costs and increase the rate at which software can be localised must be given high priority. Computer automation is an area which provides great promise in this search for reducing costs and increasing capacity. Automating the localisation process is a difficult undertaking due to the inherent complexity encountered during the localisation process. Despite this, technologies such as Translation Memory and Machine Translation represent a major leap forward in improving the rate at which software can be translated as part of localisation. Significantly, however, they are viewed mostly as enabling technologies and will not replace the human translator in most cases. This means that, the ability to acquire and manage an extremely large pool of human translators/post-editors, is a major stumbling block in improving the rate at which software can be localised.

Paper one in this series described an architecture created by Exton et al. (2009) which appears to be well placed to tackle the issue of how to acquire and manage a large pool of translators effectively. The model itself will enable the users of an application to translate the interface while they use it. The user could make a change to UI text, for instance by ctrl-clicking on the text. A popup would then appear allowing the user to update the text. The translations made by the users would then be sent back to a central server where they would be aggregated and analysed. Other users translating the same elements could view the translations from the community members. Each translated element would have an associated rating based on whether or not other community members agreed with it. At this point, there were a number of possibilities as to how the central server can choose to roll-out updates for the software. One possibility proposed by Exton el al. (2009) involved having the Crowd Sourced translations reviewed by a 'trusted' translator. Essentially, this could correspond to post-editing. Once the translations were approved, a new release could be rolled out which would update the UI elements for all the application users.

This Micro-Crowdsourcing architecture could significantly reduce cost and increase the capacity within the translation element of the localisation process. For this reason, the authors created an implementation based on this architecture. This paper describes the implementation, which has the following aims:

- To investigate the feasibility of implementing this type of architecture.
- To investigate whether Micro-Crowdsourcing can be easily applied to existing applications.
- To explore the implementation of tools which could assist in the management of the 'crowds' localisation activities (Stretch goal)
- To identify key areas which future implementations should address.

The implementation undertaken for the purpose of this paper is called Babel Software. The name for the implementation is taken from the ancient city of Babel in the land of Shinar in which the building of a tower (Tower of Babel) intended to reach heaven, was dedicated to the glory of man. God, displeased with the builders' intent, came down and confused the people's languages and scattered the people throughout the earth. Babel represents confusion and an inability to communicate. It is hoped that Babel Software can provide a means from which to minimise this confusion, by providing toolkits and services, which can help bridge the digital divide.

The Babel Server implementation as described here, does take a slightly different view than that put forward by Exton et al. (2009). Although this is software which could be used by commercial organisations to reduce localisation costs, the implementation of the Babel Server is designed specifically with the open source community in mind. The vision is to have a central server which could be used by a large number of open source communities to manage their localisation efforts free of charge.

Babel Software can be broken down into two distinct areas, the Babel Libraries and the Babel Server. The Babel client libraries provide localisation tools which program developers can easily incorporate into their applications. These tools will enable users to localise content from within the application. The Babel Libraries can submit translations to a central server as well as request a list of recommended translations. The Babel Server acts as the central hub to co-ordinate localisation efforts by servicing client requests. The Babel Server also provides essential tools to manage individual projects.

The Babel Libraries will be utilised by projects wishing to localise their software using Micro-Crowdsourcing. Note that throughout this paper, applications which use the Babel Library will be referred to as the Client. From the client's perspective, once they have included the Babel Library as part of their implementation and the software has been internationalised, the localisation process can now begin using Micro-Crowdsourcing. The client owner can then register the project on the Babel Server. Once this is done the client application can be distributed to a community of users by the client owner. Note that the client application will also be available on the Babel Server website. Once at this stage, users operating the application can translate various parts of the application as they use it. The community translators may be working with a base English translation for example or possibly a rough pre-translation using Machine Translation and/or Translation Memory. These translations are then shared with the rest of the community via the Babel Server. Once all elements have been translated the project owner can bundle the translations as part of the applications final release.

The next section focuses on describing the architecture of the Babel Client and the functionality it provides. This is followed by a walkthrough of the functionality provided by the Babel Server and its architecture. Then some of the more significant findings of the Babel Software Project will be discussed. This will be followed by a detailed description of the required future work. Finally the paper will conclude with some final thoughts.

Babel Library

Currently, Babel Software contains a single implementation of a Babel Library for Java Swing Desktop Applications. It is hoped that in the future, additional implementations will be created to cater for other languages. This chapter focuses on describing the architecture and functionality of the Java Babel Client which is called JBabelLibrary.

It is important to note that the Babel Library is not a standalone application; rather it provides functionality which other applications can use. To facilitate the development of this project, an open source application called 'Rachota TimeTracker' was customised to work with the Babel Library. This is a portable java application which is used for time-tracking projects. Again, please note that throughout this paper, applications which use the Babel Library will be referred to as the Client.

The architecture diagram in figure 1 is useful as an introduction to the Babel Library and places it in context. The functionality that the Babel Library provides is utilised by a client, such as Rachota TimeTracker. The Babel Library in-turn performs actions on relevant language files on the operating system and communicates with a larger community of translators via the Babel Server.

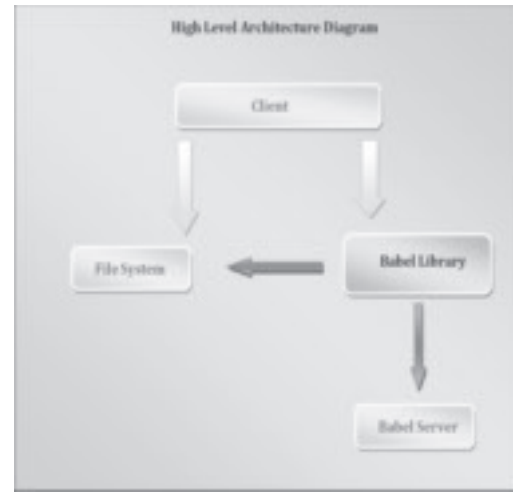


Figure 1: High Level Diagram

Layer 1: Babel Components

A major goal throughout the design of the Babel Library was to make it extremely easy for developers to utilise the library, to the point of adding little or no extra effort to the development process when compared to traditional development. The diagram below is designed to illustrate this point. Developers for instance, normally create JButton's to display clickable buttons on their user interfaces. To implement the Babel Library, developers would

instead instantiate JBabelButton's. The JBabelButton class sub-classes JButton and therefore acts in the very same way as a JButton except it provides in-built localisation tools. There are seven Babel components in total, each one sub-classing a different java class. The Babel Library contains many layers which will be described in this section; the Babel Components layer shown in figure 2, is just one such layer. For the most part clients which utilise the library only need to interact with this layer and need not concern themselves with the inner workings of the library.

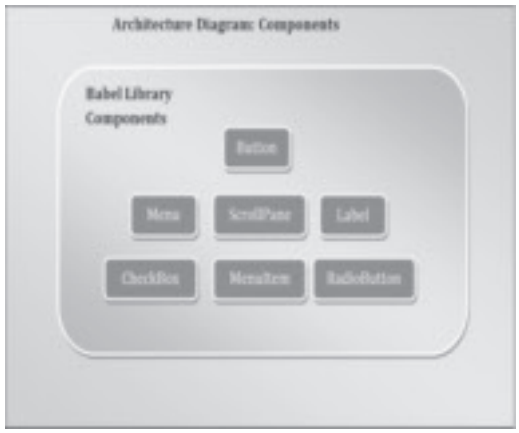


Figure 2: Babel Components (first layer)

Babel Component List

The table below contains a list of the Babel Components and their parent Java Class.

Babel Implementation	Java Class
JBabelButton	JButton
JBabelMenu	JMenu
JBabelScrollPane	JScrollPane
JBabelCheckBox	JCheckBox
JBabelLabel	JLabel
JBabelMenuItem	JMenuItem
JBabelRadioButton	JRadioButton

Table 1: list of the Babel Components and their parent Java Class

Example

Example 1 shows how a client application can implement a Babel button. The JBabelButton is still of type JButton and will behave in much the same manner as the parent class. Each Babel Component has a number of constructors depending on what parts of the component are to be localised. In this example, there is one parameter indicating that it is the primary text within the component which can be localised by the BabelLibrary. In the Babel Software Project, the client application, Rachota TimeTracker, had a language file associated with it on the file system. The parameter passed into the constructor is used to find and translate the correct value within this language file. The parameter is also passed to the Babel Server when requesting and submitting translations. A more detailed explanation of the role of the parameter(s) in the constructors will follow in subsequent sections.

```
JButton exitButton = new JBabelButton("key.exitButton");
```

Example 1: Implementing a Babel button

Each Babel Component has a number of bespoke constructors. Depending on what elements within a component require localisation, a different constructor is called. Currently all components contain two types of constructors.

- Components requiring their main text to be translated.
- Components requiring their main text and tool tip text to be translated.

These constructors contain code which fires events when a component is right clicked.

Layer 2: Babel Interfaces

This sub-section will describe the implementation of the Babel Components and their interaction with the next layer in the hierarchy called Babel Interfaces.

The bespoke constructors contained in the Babel Components provide the link between the first and second layers of the library. Once a Babel Component is right clicked, a call is made to the Interface BabelPopUp. A reference to the calling component itself is passed in the method call.

The BabelPopUp is essentially a popup menu. The popup menu affords the user the opportunity to enter

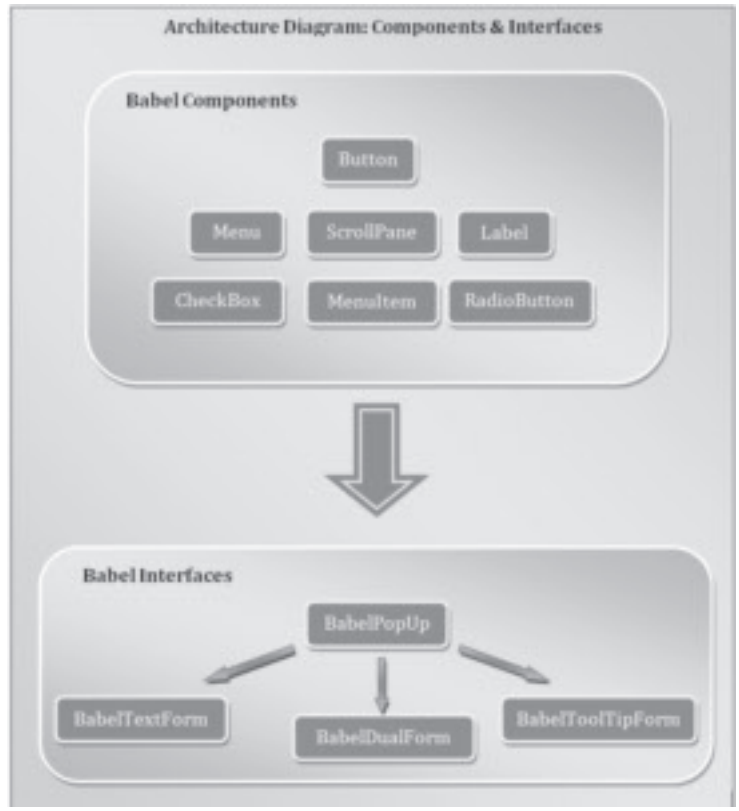


Figure 3: Babel Components & Interfaces (Layers 1 & 2)

a Babel Translation menu. This is illustrated in figure 4: the Rachota Timetracker application has been edited to incorporate the JBabelLibrary. All the buttons labels and menu items seen here, within Rachota, are actually Babel Components. In this instance, the user has right clicked on the button at the top left corner and is presented with a popup. The popup displays an option to 'translate' the component.

Once the user selects 'Translate', the Babel Popup menu disappears and is replaced by one of the Babel Forms. The Popup menu class is responsible for deciding which form is to be displayed; ultimately, this determination is made based on which type of constructor was called when creating the Babel Component.

There are three Babel Forms:

- Babel Text Form, suitable for components which contain only their primary text to be translated.
- Babel Tool Tip Form, suitable for components which contain only tooltip text to be translated.
- Babel Dual Form, suitable for components

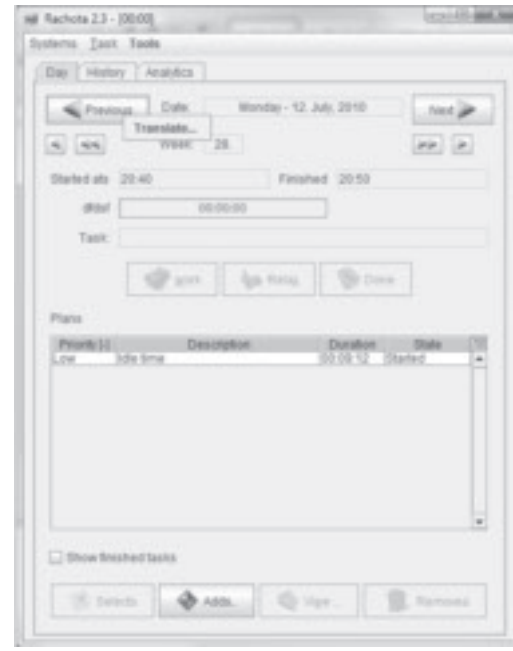


Figure 4: BabelPopUp

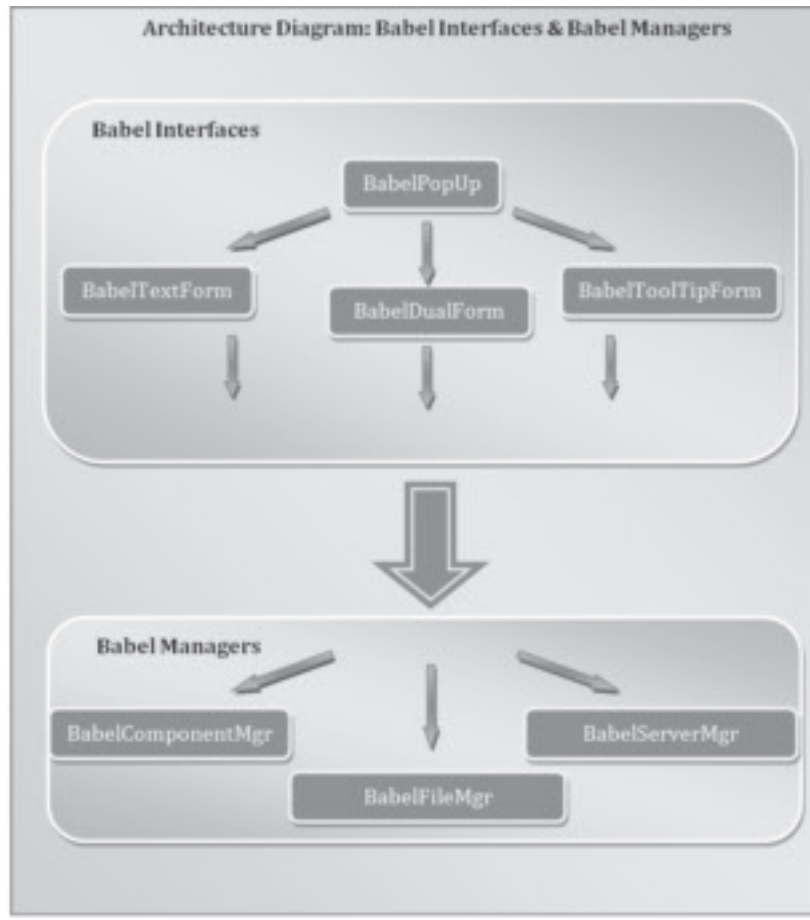


Figure 5: Babel Interfaces & Babel Mgrs (layers 2 & 3)

which contain both their primary text and tooltip to be translated.

Figure 6 shows a screenshot of the Dual Form. As previously stated the form allows translation of both a components primary text and tooltip text. The top half of the form is responsible for primary text translation and the bottom half the tooltip text translation. Both sections have exactly the same functionality.

The sections are simple in their design, allowing the user to trigger two events:

- Firstly, if the user clicks the Retrieve button, the list box to the left displays a list of Babel Community translation suggestions. These translations have an associated rating based on whether or not other translators agree with the suggestion. The Babel Form delegates the responsibility of retrieving the data from the Babel Community to the BabelServerMgr.

Figure 6: Babel Dual Form

- Secondly, there is an option to allow the user to translate the text. The user can input the updated translation and click the translate button. Once this event is fired, the Babel Form

delegates to all three Babel Managers to perform tasks such as updating the User Interface to display the updated translation, writing the updated translation to the language file and finally sending the updated translation to the Babel Community.

The tasks that the Babel Manager performs will be discussed in greater detail in the next section.

Layer 3: Babel Managers & Resources

The three Babel Managers contain the core of the functionality which the Babel Library provides. The managers are responsible for performing tasks on the File System, Application Components & Babel Server.

Babel File Manager (BabelFileMgr)

Clients that run the Babel Library must keep a language file on the system which is used to load text. It is displayed to the user in their selected language. This file and its values are typically loaded when the application starts up. In most cases, the client will store the loaded file in memory as a PropertyResourceBundle for easy access.

When a user submits a translation to the Babel Library, this file must be updated if the changes are to take effect. The Babel File Manager is responsible for this. Once the File Manager has updated the language file, it raises a flag to show that the File has been updated. This prompts the client application to update its PropertyResourceBundle in memory.

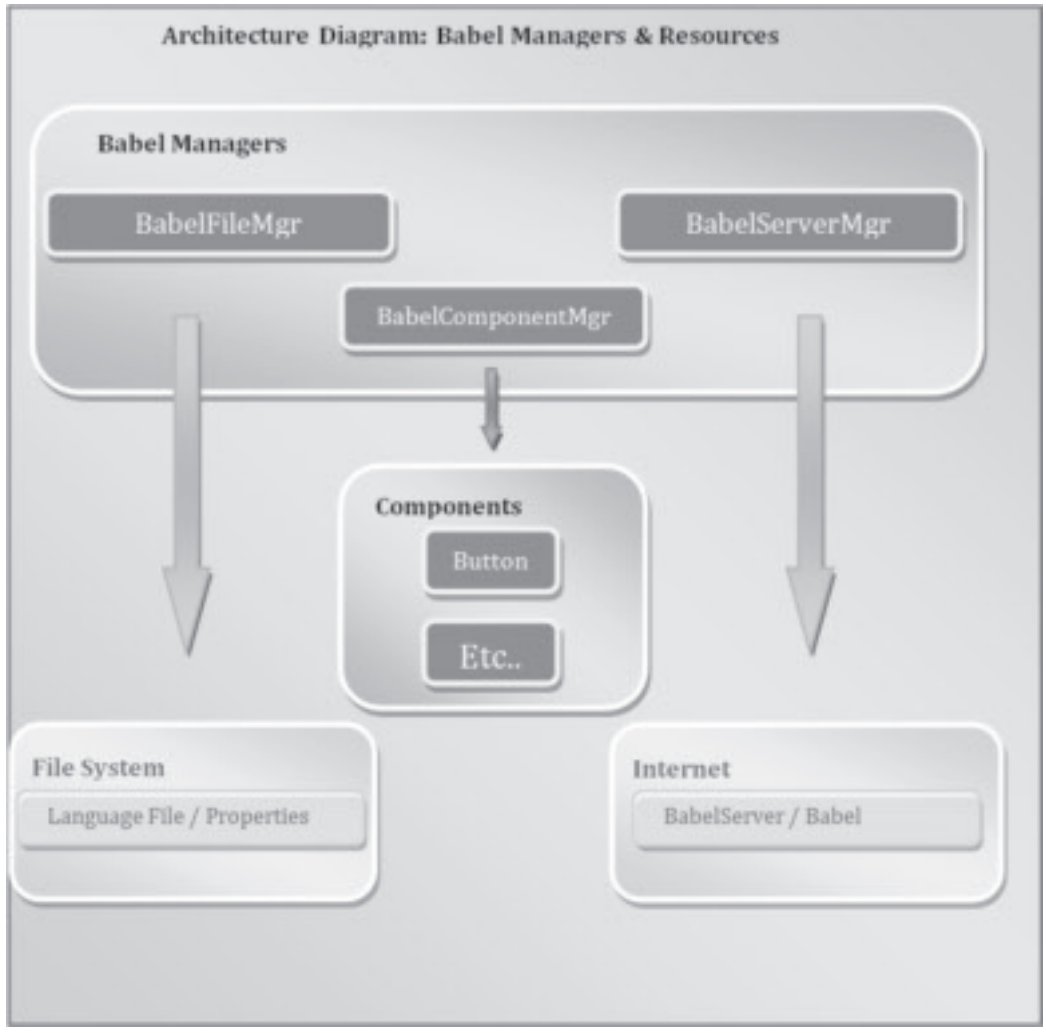


Figure 7: Babel Managers & Resources

Babel Component Manager (BabelComponentMgr)

The File Manager updates text of newly created components and persists the translation(s), however, it does not cater for components which have already been created. When a user has translated a component, this component is viewed as having already been created and its text must be updated dynamically to reflect the user's changes. It is the function of the Component Manager to update existing components.

The only way the Component Manager can possibly update a component is if it has been passed a reference to the object in question. When an event is fired on a component, it passes a reference to the Babel Forms which is forwarded to the Component Manager.

The Component Manager determines what type of object/component it has been passed, i.e. whether it is a JBabelButton or JCheckBox for example. Then the component's text property is updated accordingly.

Babel Server Manager (BabelServerMgr)

Central to the Babel Software is the Babel Server and its ability to co-ordinate and share localisation resources among translators. The Server persists and rates user's translations, while making these translations available to other users. The next section will discuss the Babel Server in greater detail.

The Babel Server Manager is responsible for communicating with the Babel Server. The Babel Server Manager sends data to the server via a HTTP request. Significantly, although the library is in Java, the objects it sends in the request and the objects which are returned from the Server are not Java objects. Instead, both use a language independent data interchange format called JSON. This ensures that the Server will be able to communicate easily with other libraries developed in the future.

The Babel Server Manager performs two key functions for the library:

- Firstly, it can request recommended translations from the Babel Server for a particular component. This function forwards the application name, application language and the component identifier to the Babel Server. The Server then returns a list of recommended translations with an associated rating.
- Secondly, it sends user translations to the server. It forwards the application name, application language, the component identifier and the user's translation to the Server. The Server then responds with a message stating whether or not the translation was processed.

Babel Server

This section describes the design, implementation and functionality offered by the Babel Server. The server is central to the concept proposed by Exton et al. (2009). It is the hub which co-ordinates the localisation activities of a community of translators.

Figure 8 details a high level architecture to place the Babel Server in context, relative to external applications with which it interacts. On the left side of the diagram, a client using the Babel Library is seen interacting with the server. This could represent the Rachota application as seen previously, or any other application implementing the Babel Library. Another significant facet to the Babel Server, which can be seen on the right side of the diagram, is the Babel Website. This website can be viewed like any other, via a web-browser. Its primary aim is to provide online tools from which to monitor and manage the localisation of a project/application.

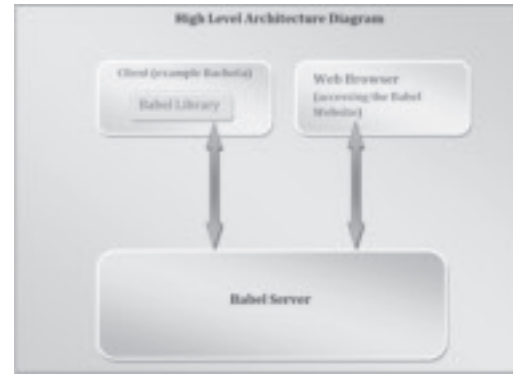


Figure 8: High Level Architecture Diagram

As can be seen from Figure 8 the Babel Server can be divided into two distinct areas, functionality that services Clients or requests from Babel Libraries and functionality which supports the website. Both sets of functionality will be described in detail later in this section. Firstly, however, a technology overview for the entire server will be presented as both areas of the server leverage much of the same technologies.

Technology Overview

The Babel Server is essentially a web server which utilises the Spring Framework. Irrespective of whether or not a request comes from a client using the Babel Library or whether it is a request to view the website, the architecture outlined in figure 8 processes the request. The Spring Framework was chosen for the Babel Server as it provides enormous flexibility. This is extremely important when keeping in mind that it is hoped that there will be future releases of the server which will build upon the existing implementation. Without the implementation of the Framework, the process of reusing or even comprehending existing code would be made extremely difficult.

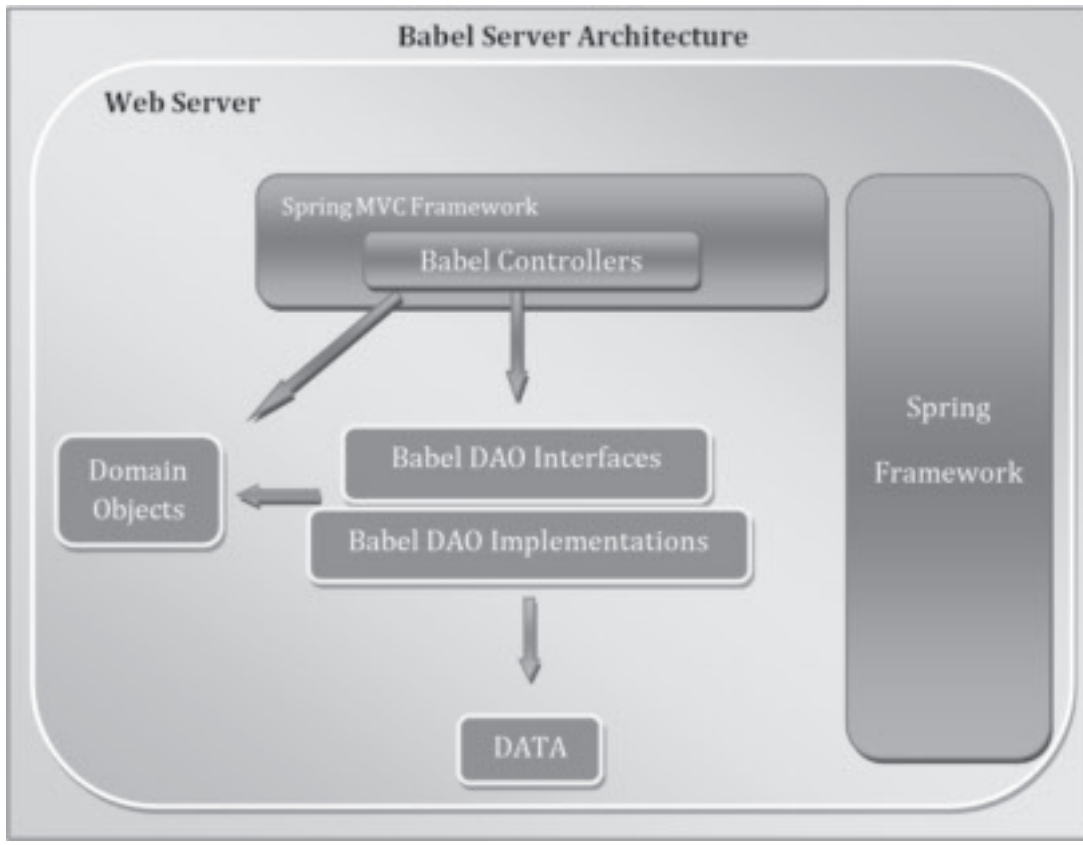


Figure 9: Babel Server Architecture

Spring Framework

The Spring Framework consists of 7 modules which cater for various aspects of JEE application development. Currently, the Babel Server implements only two Spring modules, however it is hoped that this number will increase in the future. The two modules which the server uses are; Springs Core Container and its MVC Framework. The core container is the primary module within the framework. The core container provides a means to separate configuration settings and dependencies from the code itself. It does this using the Inversion of Control (IOC) pattern. The Inversion of Control pattern or dependency injection, allows you to describe how objects are to be created without creating them and their dependencies directly in application code. This involves specifying the dependencies between objects or components in a configuration file. The container then interprets this specification and is responsible for the creation of objects and their dependencies. (Balani 2005)

The Inversion of Control pattern provides a means of separating application logic into layers which can be interchanged and reused if necessary. As mentioned earlier, this flexibility was the primary reason for

choosing the Spring Framework. An example of this flexibility in practice can be seen with the Data Access Object layer (DAO) as seen in figure 8. Although the surrounding layers are much more mature, the development of the DAO layer is in its preliminary stages, the existing implementation will be replaced in future releases. The IOC pattern enables the developer(s) to insert a new DAO, without affecting other layers of the application.

The Spring MVC Framework is a module for building web applications. Spring MVC (model-view-controller) uses the MVC architectural pattern which separates business, navigation and presentation logic (Springdeveloper.com n.d.):

- *Controller:* Contains navigation logic and interacts with other application layers like the Data Access Object (DAO) to retrieve business logic. This is illustrated in Figure 8, the Babel Controllers interact with the Babel DAO and domain objects to retrieve business logic.
- *Model:* Contains the data which is required by the View. This is retrieved from the Babel DAO.
- *View:* Is responsible for pulling data from the model and rendering the output. There are two types of views used by the Babel Server, JSON and JSP. JSON is what is returned to clients using the Babel Library and a JSP is returned when accessing the Babel website.

The key benefit to Spring MVC is similar to that of its parent. The separation of logic makes the code much easier to change in the future.

Data Structure

As can be seen in Figure 8 the controllers and DAO utilise the Domain objects. The Babel implementation contains three Domain objects, Project, Dictionary and DictionaryElement. The data structure is designed to capture the various types of data required to describe the communities' translations for all projects:

- An object of type Project could represent an application such as Rachota or Open Office. The class contains simple data types to store the project name, date created and authorisation data. Also, each project can have a list of Dictionaries associated with it.
- Each language that an application caters for would be encapsulated within a dictionary object. The dictionary class contains information on the language/locale the dictionary represents as well as a list of dictionary elements. The dictionary class also provides functionality to search for dictionary elements and update specific element attributes.
- Dictionary elements represent components, such as buttons and text fields in the client application. Each dictionary element contains a number of fields, which store the components unique identifier used by the client application, as well as an associated translation. Significantly, the translation is also given a rating based on the community's approval/disapproval.

Rating system

Contained within the domain objects is the logic for the voting system. The current implementation enforces a very simple system, whereby if another community member agrees with a translation, its associated rating is simply incremented.

Library requests

This sub-section describes the implementation of components which service Babel Library requests. The diagram in figure 9, identifies the main areas of interest and the flow of control with regard to servicing library requests. For simplicity, the diagram below shows only one client library. In reality the server can process requests from multiple client applications.

One of the most important aspects of the diagram below is the fact that the server returns JSON objects to the client. JSON is a lightweight data interchange language; the rationale for choosing this format is closely linked to the future needs of the Babel project. If the development of no other client libraries was envisaged then the server could simply return a Java object which could be easily understood by the client. However other client libraries are expected to be developed which cater for other languages. Therefore returning language specific objects is not appropriate. Using JSON means that the server does not need to change in order to accommodate other client libraries and in turn all client libraries can easily parse responses from the server.

Client applications implementing the Babel Library can request two services from the Babel Server:

- Request a list of recommended translations
- Submit a translation.

The client can request these services by making a

HTTP request and passing parameters including application name and the name of the component to be translated. The web server then directs the Babel Library request to the relevant Spring MVC controller, either the 'requestTranslationController' or the 'submitTranslationController'. The controller then retrieves the 'Model' or the data by delegating to the Data Access Object (see Figure 8). Once the Model has been retrieved the Controller directs the flow of control to the JSON view (see Figure 9).

The Spring Framework requires that all information sent back to the client is sent via a class implementing the Spring Interface 'View'. Although spring bundles implementations for JSP's and XML for example, it does not cater for JSON. Therefore the JSON View and JSON Util classes shown in figure 8 were created during the development process. The JSON View implements the Spring Interface View. It has an associated helper class called 'JSONUtil' which converts the data to a JSON

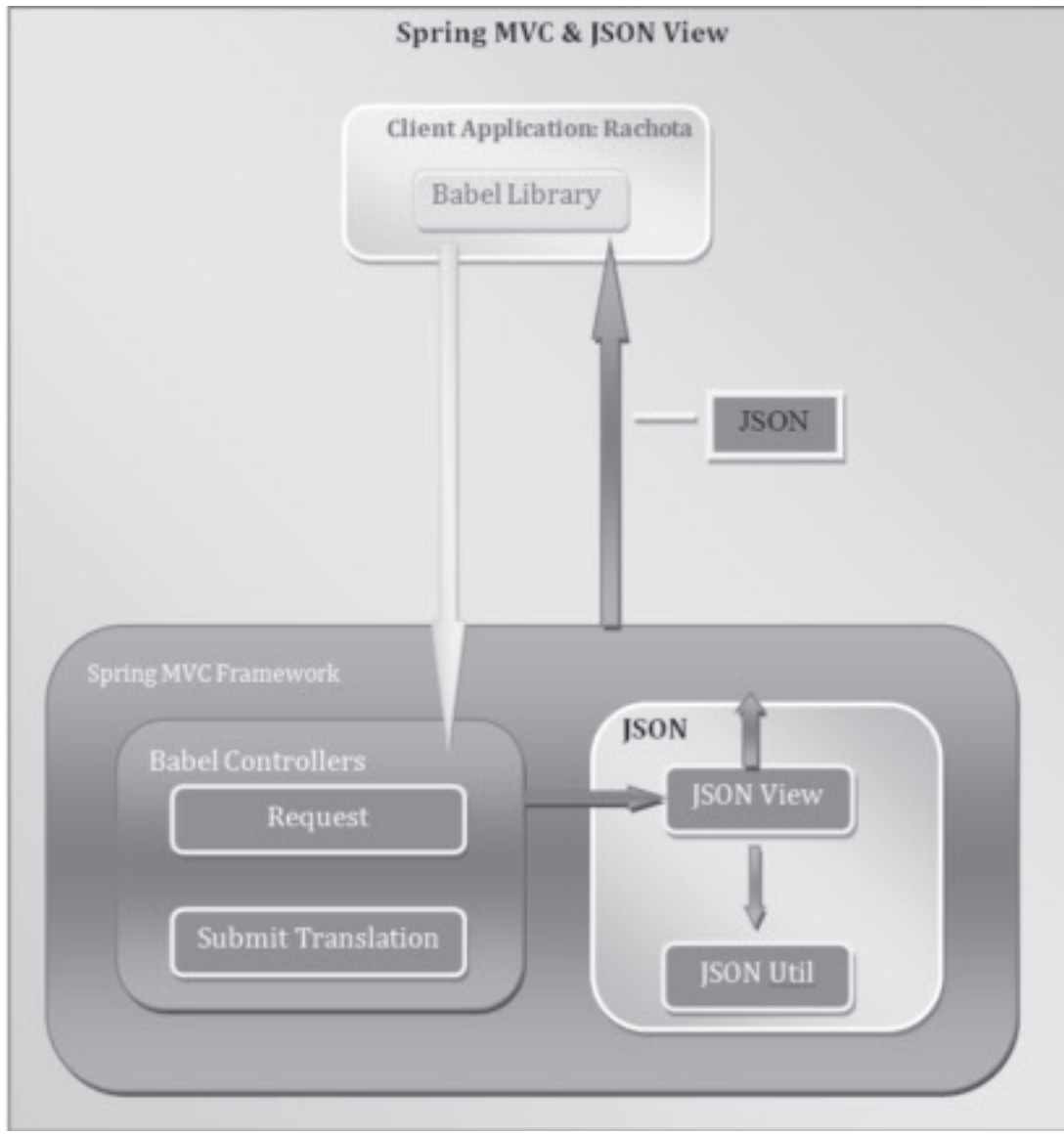


Figure 10: Spring MVC & JSON View

object. The JSON view then forwards the JSON object to the client.

Babel Website

The Babel website has two major driving factors: Firstly the website provides a valuable means through which to advertise Babel Software and subsequently encourage users to join the community. Secondly, the website provides tools to manage projects running on the server. This sub-section begins with a brief description of its implementation, followed by a description of the functionality provided by the Babel Website.

Architecture

The processing of requests from web browsers is much the same as requests from client libraries. There are five Babel Controllers including 'Add Dictionary Controller' and 'Delete Dictionary controller'. The controllers retrieve the 'Model' or the data by delegating to the Babel DAO *(see Figure 8)*. The Babel DAO performs operations such as adding or deleting dictionaries associated with a given project. Once the Model has been retrieved, the Controller instantiates a View object. This process differs from the previous section as a JSP View is used instead of a JSON view. As stated previously Spring does bundle a JSP View with the Framework making this process less time consuming. During the instantiation of the JSP View, it is attached to a specific JSP page such as 'addDictionary.jsp' or deleteDictionary.jsp *(see Figure 10)*. These JSP's are then displayed in the client's web browser as a web page.

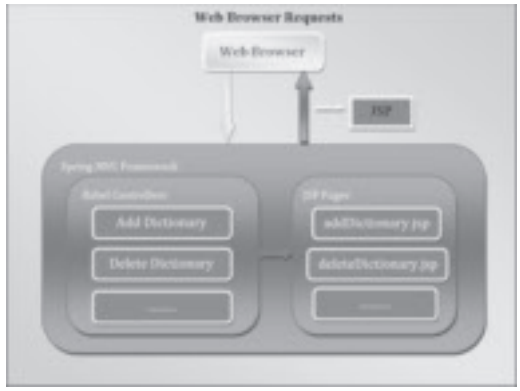


Figure 11: Web Browser Requests

Advertising the Babel Community

The first page or homepage of the website is largely targeted at engineers. It provides a brief overview of what Babel Software is and provides links to resources such as how to configure the Babel Library and manage a Babel account.

Figure 12: Website Homepage

The Babel Download centre is easily accessible from the homepage. It provides an option to download the Babel Client library (currently only for Java Swing). Also there is a link to the source code which directs the user the Babel Software SourceForge page. Both the Babel Client and Babel Server source code is open-source and is accessible via a subversion repository.

Figure 13: Website Download Page

Project Management/Administration

The Babel Server implementation for the website allows a project administrator for a project such as Rachota to login and manage and oversee various aspects of the community. Figure 14 shows how users can log into the site and enter the Administration area for their project.

Figure 14: Website Login page

Figure 15: Website Administration area

Figure 15 is a screenshot of the Administration area for an administrator of the Rachota application which was mentioned earlier. In this sub-section we will describe three of the options available to the administrator:

- Add a dictionary
- View Translations/dictionary
- Delete a dictionary.

Figure 16 shows a screenshot of the add dictionary page. In this instance the project administrator is attempting to create a German dictionary for the Rachota application. From here the project administrator can upload a base language properties file. The uploaded properties file could, for example, contain rough machine translations for the language in question. Once uploaded the Add DictionaryController' delegates to the DAO which creates a new dictionary for the project as well as dictionary elements based on the data in the properties file. Note that each element uploaded in the properties file has its rating set to 0. Once this process is complete, clients with the German version of Rachota can submit and request translations to the Babel Server.

Figure 16: Add Dictionary Page

Throughout this process, the project administrator can view the communities' progress and can see ratings for the various dictionary elements as seen in figure 17. When the View Dictionary page is requested by the user the 'ViewDictionaryController' delegates to the DAO which retrieves data in the form of domain objects to be displayed to the user. The dictionary information displayed to the user could be an extremely valuable tool in overseeing the translation process.

Figure 17: View Dictionary

The final page in the website walkthrough is the delete dictionary page. This allows the project administrator to remove a dictionary from the server by simply clicking on the red button as shown in figure 18.

Figure 18: Delete dictionary page

Discussion

The implementation of the Babel Client Library and Babel Server as described in the previous two sections was deemed to be a success. No significant road blocks were encountered and a great deal of functionality was implemented for the Client Library and Server.

This section will discuss some of the more significant findings and experiences during the implementation process which relate to three of the four goals of this exploratory implementation:

- To investigate the feasibility of implementing this type of architecture.
- To investigate whether Micro-Crowdsourcing can be easily applied to existing applications.
- To explore the implementation of tools which could assist in the management of the 'crowd's' localisation activities (Stretch goal)

Client Library

The implementation of the Babel Client Library for Java was an extremely time consuming process. The implementation of the babel.client.core package (see figure 19) was the most demanding and was of critical importance to the success of the project. In contrast, the implementation of the components package was much less demanding. Therefore, based on the development of the seven Babel components, the expansion of the library itself to include other components should be possible in a shorter timeframe than first expected.

One area where issues were encountered was when attempting to retro-fit an existing application with the Library, as opposed to building an application with the Babel Library in mind. Significant changes were made to the client application Rachota, to facilitate the Babel Library.

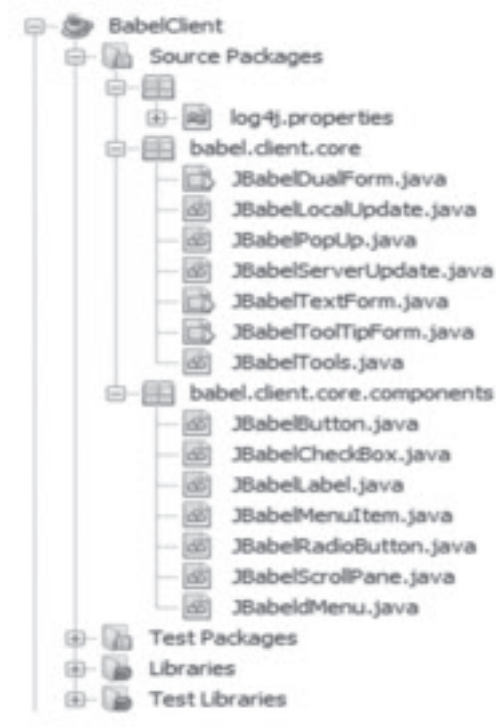


Figure 19: Babel Client Packages & Classes

As is standard with Java Swing applications, Rachota is contained in a jar file which is an executable. Clicking on this file starts the application. On start-up Rachota reads from a properties file which contains translations for a given language. The language file that the application calls is dependent on the locale settings of the operating system on which the application is run. Once Rachota has read in the file it loads the data into a PropertyResourceBundle. Using the PropertyResourceBundle makes it easier for the application to query the data which it read from the properties file. When rendering each screen to the user, Rachota calls on a class called Translator to populate values for parts of the application that have been internationalised.

```
Translator.getTranslation("DAYVIEW.BT_NEXT")
```

Example 2: Rachota calling the Translator class

The translator class queries the Property ResourceBundle in memory, for the correct translation. This implementation presented a number of issues to the development process. Firstly, the properties files or language files are kept within the

executable/jar file. This means that the 'Babel File Manager' cannot update the file with translations from the user or other members of the community. Therefore, the language files must be kept outside the jar file so that they can be written to as needed. Rachota code was updated to reflect this. Secondly, when retrieving a translation for an element, Rachota retrieves the translation from the ResourceBundle and not the file itself. Therefore, if any change has been made to the file since the ResourceBundle was created, then the ResourceBundle will not contain up to date information. In order to compensate for this, the Babel Library flags when any changes have been made to the properties file. Subsequently, a Rachota method called Translator.getTranslation(..) was altered; before it reads from the ResourceBundle, it checks to make sure that it is up to date. If the ResourceBundle is not up to date, then Rachota clears all data in the ResourceBundle and re-populates the object with data from the updated properties file. There were a number of other minor issues which had to be overcome with Rachota when trying to utilise the Babel Libraries functionality; however, once these issues had been overcome, it was found that Babel components can be incorporated quite easily into the implementation of a client application. This involves the replacement of code seen below for a JButton with the instantiation of a JBabelButton:

Example 3: Original Rachota JButton code

Example 4: Modified Rachota JBabelButton code

The text in the constructor for the Babel Component was used to locate the relevant translation in the properties file. Calling the setText simply populates the text of the component with the appropriate translation.

The difficulties faced incorporating the library into the Client application Rachota were not what one might describe as difficult engineering activities, however, they were very time consuming. When keeping in mind that the code base for Rachota is quite small and its internationalisation classes are relatively concise, one can imagine that attempting the same activity with a larger application could be a major engineering activity indeed. This would suggest that the incorporation of the Babel Library into a client application should be done during its development process. At the very least, the design of its internationalisation classes should be conceived with a view to accommodating the Babel Library. This step alone need not necessarily add to the initial development cost and would ensure that the client would be compatible with the Babel Library.

Server

Throughout the development process, the Babel Client implementation was given highest priority. Therefore, the Server was given a shorter timeframe for development than the client.

The primary goal for the development of the Babel Server was to create a skeleton implementation which, firstly, could service client requests from Babel Libraries written in different languages and also serve as the base implementation from which future releases would build upon. Therefore, great thought was given to the overall design of the server and its constituent parts. As was mentioned earlier, the Spring Framework was used in the implementation of the Babel Server; although for smaller projects or once off implementations, the use of such a framework could be seen as needless and labour intensive. Due to the advantages provided by techniques, such as dependency injection, it was seen as essential in ensuring that the current implementation could be built upon in the future. The use of an intermediate data interchange format, via HTTP protocol, is the key to ensuring that the server is compatible with other client library implementations. The concrete classes were then 'wired' together using Spring. Throughout this process, various revisions of classes were introduced and subsequently re-wired into the framework with ease, further highlighting the importance of the framework even at this early stage.

Figure 20 contains a screenshot of the package and class structure of the server. The large number of classes and files created for the Babel Server reflects the high level of productivity during this phase of development. As a result, the development of the Babel Server went far beyond the initial goal of building merely a base implementation. Instead, the server provided a testing ground for ideas and aided

Figure 20: Babel Server Packages & Classes

in the discovery of new possibilities. The idea behind Babel Software is that a small community of translators work together to localise a piece of software. The product owner has now been unburdened from a great deal of the 'heavy lifting' associated with this task. Also, the rating system provides an autonomous way of collating and grading the contributions of translators. All of this is key in reducing the cost of localisation. However, this does not mean that a product owner can simply ask a community to translate a product and come back after a few weeks and hope to find a complete translation. The product owner must administer or manage the translation process. The speedy development process, facilitated the exploration of the functionality which the server could and should provide in the future. The functionality which the server should provide, was derived from the vision of creating a Babel Server which would service a large number of open-source projects free of charge, similar to SourceForge.net. As stated in the introduction to this paper, although supporting open-source communities is the primary aim of this development effort, it could also be leveraged by commercial organisations. With this in mind, the development of a website to promote the service, and the development of an administration area containing a toolkit were the focus for this exploratory work.

The vast majority of this exploratory work surrounded the development of the administration area. The ability for a project administrator to log in to the system and perform various management tasks undoubtedly is a very powerful capability. Even during initial testing it was obvious that the ability to add a new dictionary on the fly was enormous. Within seconds the infrastructure could be put in place to support the work of a group of translators. Once in place, the ability to monitor the communities can help gauge the project's progress as a whole and view the quality of translations. This monitoring facility could be an ideal tool for language experts; and could serve as the basis for future implementations which support auditing or approvals of the community's translations.

The aforementioned experiences during the development process for this exploratory implementation have led the authors to following assertions: Firstly a full implementation of the Micro Crowdsourcing architecture appears to be a feasible undertaking. Significantly, the design of the exploratory implementation lends itself to being the primary building block for this full implementation. Also, the author's experiences suggest that attempting to retro-fit existing applications with this Micro-Crowdsourcing architecture could be a time consuming and expensive undertaking. Finally, the exploration of possible management tools for use on the Server by an administrator suggests that they could provide an extremely powerful solution without incurring significant development costs.

Conclusion

Babel Software is an exploratory implementation of a Micro-Crowdsourcing architecture proposed by Exton et al. (2009). The Babel Software Project is an Open source implementation which can be found on SourceForge.net. Babel Software consists of two main parts, the development of a client library called Babel Library and a server called the Babel Server. As part of the development of the Babel Library, it was incorporated into an existing Java Desktop application or client. The development of the Babel Client Library and Babel Server prototypes proved successful. The development process was demanding yet found that the development of the Micro-Crowdsourcing architecture was feasible. It also found that, although it was possible to retro fit an existing application with the Babel Client Library, it was time consuming and not advisable. Therefore, any client application wishing to leverage the Micro-Crowdsourcing architecture should incorporate the Babel Library into its design during the initial development phase. At the very least, the design of a clients internationalisation classes should be conceived with a view to accommodating the Babel Library. A stretch goal of the implementation was to develop management tools for users on the Babel Server. This proved to be an extremely important aspect of the implementation and helped provide great insight into the infrastructure which the Babel Server should provide in future releases.

The implementation of this Micro-Crowdsourcing architecture, has the capability to increase the volume of localised material without necessarily compromising on quality. Although it is not seen as a complete solution to meet the increasing demand for localisation services, it is well placed to complement other technologies in the localisation workflow. It is hoped that technologies such as Machine Translation and Translation Memories together with Micro Crowdsourcing, can combine to dramatically decrease costs and increase the rate at which software can be localised.

References

Asnes, A., 2009. Internationalization ROI. MultiLingual, 20(6), 32-35.

Balani, N., 2005. The Spring series, Part 1: Introduction to the Spring framework. Available at: http://www.ibm.com /developerworks/web/library/wa-spring1/.

Collins R.. 2001. Software Localization: Issues and Methods, The 9th European Conference on Information Systems. Bled, Slovenia.

Ebben, S. and Marshall G. (1999) The Localisation process: Globalizing your Code and Localizing your Site. http://msdn.microsoft.com/workshop/management/intl/locprocess.asp (accessed on 6th June 2000)

Elliott, J., 2005. What is Hibernate. Available at: http://onjava.com/pub/a/onjava/2005/09/21/what-is-hibernate.html.

Exton, C. Wasala A., Buckley J and Schäler R., 2009. Micro Crowdsourcing : A new Model for Software Localisation. The International Journal of Localisation, 8(1), 81-89.

Fernández N, 2000 Web Site Localisation and Internationalisation: a Case Study. MSc Thesis, City University London.

MySQL, Why MySQL. Available at: www.mysql.com/why-mysql/.

Schäler, R., 2004. The Cultural Dimension in Software Localisation. Localisation Reader 2003-2004, 5-8.

Smith, B., 2007. A Quick Guide to GPLv3. Available at: http://www.gnu.org/licenses/quick-guide-gplv3.html.

Sourceforge.net, 1999. About Sourceforge. Available at: http://sourceforge.net/about.

Sourceforge.net, 1999. About Subersion. Available at: http://sourceforge.net/scm/?type=svn&group_id=332838.

Sourceforge.net, 1999. Sourceforge Homepage. Available at: http://sourceforge.net/.

Springdeveloper.com, Spring MVC. Available at: http://www.springdeveloper.com/oscon/mvc.pdf.

Guidelines for Authors

Localisation Focus
The International Journal of Localisation
Deadline for submissions for VOL 10 Issue 1 is 30 June 2011

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
- 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.

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. 9 Issue 1 (2010)

CONTENTS