Localisation - When Language, Culture and Technology Join Forces - By: JODY BYRNE
Posted by Michael Lambarena on June 04 2009 21:44:16

Localisation - When Language, Culture and Technology Join Forces

When you switch on your computer and type up a letter, what language do you see? What about when you visit a website or play a computer game? Does your mobile phone speak your language? Chances are that each of these technological marvels of the modern age communicates with you in your own language. For many of us, this is so commonplace and seamless that we hardly give it a moment's thought but behind the scenes there is a whole industry dedicated to making sure that technology bridges the gap between language and culture without you even noticing.

Once upon a time, if you wanted to use a computer for whatever reason, you had to be able speak English. The alternative was a tedious process of trial-and-error using a dictionary and your powers of deduction. The reason for this is that Personal Computers were originally developed in the sunny, English-speaking climes of Silicon Valley in the USA where engineers and programmers concerned themselves with producing the next technological break-through. Back in the 1980s it never occurred to companies that there could be people in the world who did not speak English, or worse, who, even though they spoke English, actually preferred to speak their own languages. Over time, however, companies realised that in order to break into foreign markets and maximise profits, they would have to provide foreign language versions of their software rather than expect those pesky foreigners to learn English.

And so, once software was developed it was sent back to the developers who were told to "translate" it into whatever languages were required according to the company's sales and marketing goals. Developers were less than enthusiastic about this, naturally. After all, they had done their job and now they were expected to do even more work which, strictly speaking was not their job. What's more, because individual products, like languages, had their own peculiarities, customs and conventions, the process of translating the software was often time-consuming, incredibly complex and not always successful. One way of describing this process is to imagine baking a fruit cake and then being told afterwards to remove the raisins from it!

So what is localization?

From such less than auspicious beginnings roughly 20 years ago, the computer industry has come a very long way indeed. Today it is almost inconceivable that anything other than the smallest of software companies would fail to provide different language versions of their products. This has given rise to localization, an industry estimated to be worth in the region of anywhere between $5 and 15 billion (LISA 2007) annually. But to understand localization - which is sometimes referred to as L10n, 10 representing the number of letters between the L and the n - it is important to understand that although translation plays an integral role, localization is more than simply "high-tech" translation.

Bert Esselink, one of the leading figures in localization, once described localization as "combining language and technology to produce a product that can cross cultural and language barriers" (2003:4). A similar definition is provided by the Localization Industry Standards Association (LISA) which describes localization as "the process of modifying products or services to account for differences in distinct markets" (LISA 2008). This second definition is perhaps the most useful for understanding the nature of modern localization because, rather than restricting itself to "products" which we can equate to relatively tangible things like software and DVDs etc., it includes services. This is possibly one of the most interesting developments as it sees localization moving away from being purely concerned with software and incorporating such things as websites and online gaming.

Localization however, is not an isolated process which can be undertaken without any planning. In actual fact, it is a complex activity which requires linguistic, technical, cultural, commercial and legal expertise in order to get it right. For localisation to be effective and cost-efficient it needs to take place within a larger commercial and developmental framework involving the expertise of translators, language and tools specialists, programmers, engineers, project managers, desktop publishing specialists and marketing staff (Folaron 2006). This framework can be explained using the acronym GILT. Consisting of Globalization, Internationalization, Localization and Translation, it represents a consolidated process whereby companies put in place the procedures and mechanisms needed to function effectively in a global market.

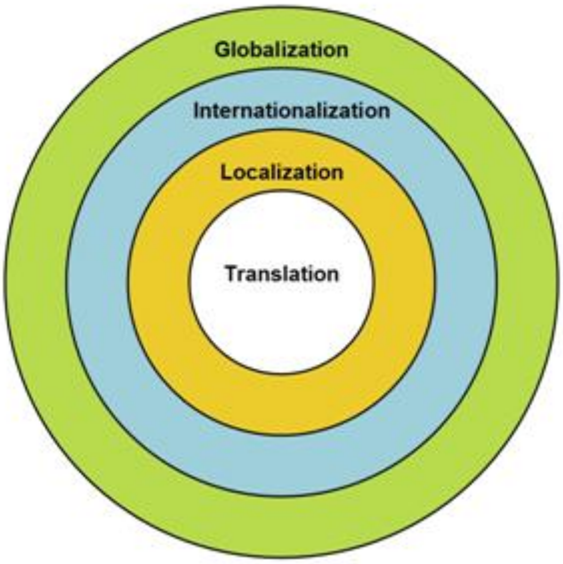


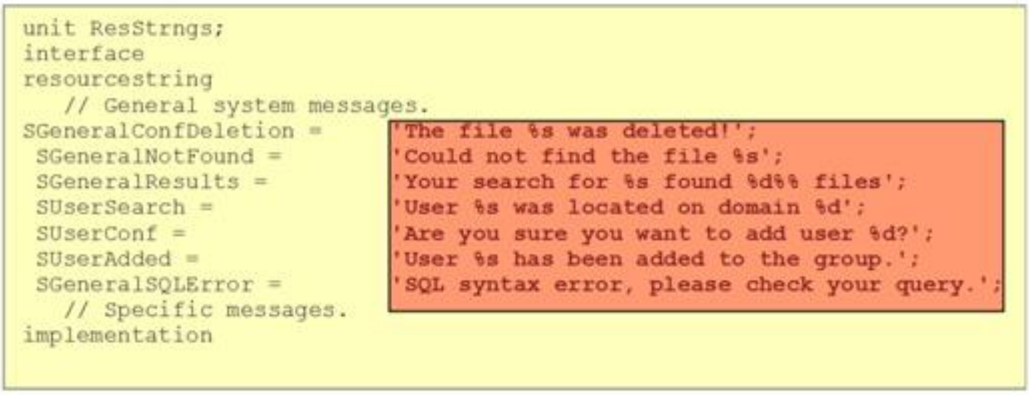
Figure 1: The GILT framework

Globalization involves making all of the necessary technical, financial, managerial, staffing and other decisions needed in order to be active in a global market and to sell products which require only minimal revision. Internationalization involves designing and engineering products in such a way that they can be localized as easily as possible. This requires a design and development process which considers the needs of all intended international users from the outset so as to avoid many of the problems and barriers to effective localization before localization even begins. Such an approach can help prevent potentially disastrous situations where, half-way during the localization process, a feature of the product's design means that localization and/or translation cannot proceed without substantial and costly re-designing of the entire product. Localization is the actual process of customising the product for use in other markets while translation is the process of turning textual information in one language into textual information in another language.

Enabling localization

There are various technical and cultural issues which need to be considered during the design and development phase to ensure that localization proceeds as smoothly as possible. The aim of this process is to design products which are as culturally neutral as possible. In other words, products should not be so entrenched in a particular culture that it cannot be adapted for use in another locale without significant engineering work. The notion of a locale is an interesting one and it nicely encapsulates many of the aims of localisation. It can be defined as a combination of a language and factors relating to the particular region or country in which that language is used. This is an important idea as it allows us to distinguish between localizing a product into Spanish for use in Spain and one for use in Mexico or Ecuador. Similarly, it allows us to differentiate between software for the US market and software for the British or Australian markets, for example. All have the same language but the different language variants combined with different cultural and legal factors mean that they are not interchangeable. In order to create products which can be slotted into different locales, a range of technical and cultural factors need to be considered. This requires not only a sound technical understanding of the technologies but a profound knowledge of the target culture and their expectations, customs and preferences.

One of the most important technical factors in internationalization is the idea of separating the machine code from the user text, typically through the use of code pages or string tables (see Esselink 1998:21-22 and Heimburg 2006). Machine code is the part of a software application (and nowadays websites) that provides instructions to the computer telling it what to do. This type of code is not of interest to users and typically does not contain translatable content. User text is text which is displayed on-screen to users in the form of menu options, error messages, dialog boxes and so on. In the old days, this type of text was scattered throughout the hundreds and sometimes thousands of files that make up a piece of software and this made localization very difficult. Not only did translators need to find the text to translate but, because it was hard-coded into the machine code, a simple spelling mistake, or the accidental deletion of something as simple as a colon or question mark could stop the entire program from working. To solve this problem, developers of internationalized products collect all of the strings of text that will be displayed to users in a relatively small number of special files known as code pages. Each text string which will be displayed in the user interface (UI) is represented by a variable - its own unique reference code - which is used throughout the program instead of the actual text. When the program needs to display a piece of text, it uses the reference code to locate the correct string in the code page which it then displays on screen. This approach makes it much easier to find and extract the text and it also helps to make sure that no text is accidentally left untranslated. What is more, since there is no need for translators to access files containing the machine code which makes the software work, the likelihood of program errors being introduced through the unintentional deletion of codes or commands is minimised.



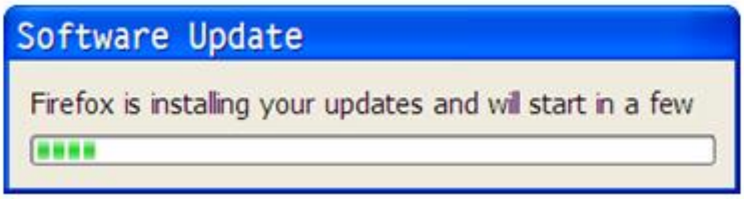
```
unit ResStrngs;
interface
resourcestring
   // General system messages.
SGeneralConfDeletion =       'The file %s was deleted!';
 SGeneralNotFound =          'Could not find the file %s';
 SGeneralResults =           'Your search for %s found %d%% files';
 SUserSearch =               'User %s was located on domain %d';
 SUserConf =                 'Are you sure you want to add user %d?';
 SUserAdded =                'User %s has been added to the group.';
 SGeneralSQLError =          'SQL syntax error, please check your query.';
   // Specific messages.
implementation
```

Figure 2: Example of text strings in a code page

Developers also need to give products the ability to handle different writing systems, special characters and punctuation. While this is less of a problem with Western European languages, it is imperative that products being sold in countries which have right-to-left writing systems (such as Hebrew) or bi-directional writing systems (such as Arabic) are capable of displaying the text for that country's language. Similarly, languages with "complex" writing systems such as Chinese, Japanese and other Asian languages for example, require twice the amount of data to describe and display individual characters. Known as "double byte scripts" these languages require products to be specifically engineered to display them.

But even indicating languages on a website or in an application can prove problematic for the unwitting designer who may be tempted to use national flags to indicate different language options. But taking English as an example we can see immediately how this can lead to problems and potentially lost customers, because it can be represented by US, UK, Irish, Australian and Canadian flags. Choosing one over the other makes a statement about the importance of the other countries and may alienate potential customers in those markets. So as tempting and visually appealing as flags may seem, their use in this way should be considered with extreme caution.

The design and layout of interfaces is also affected by a phenomenon which translators are very familiar with - texts may expand or contract depending on the language pair and direction of translation. So for example, a text translated from English into Spanish may expand by up to 30-40% due to the different grammatical structures of English and Spanish and the length of words. Consequently, interfaces need to be designed in such a way that they can accommodate any text expansion during the translation process. The following example from a well-known Internet browser shows what can happen when designers do not provide adequate space in an interface for text.


Software Update

Firefox is installing your updates and will start in a few

Ideally, interfaces should be designed in such a way that any expansion or contraction will not affect the layout of positioning of other interface elements. In the following example, the layout leaves virtually no free space to accommodate text expansion and the end of the German word "Verzeichnis" has been cut off. Another problem with this type of layout is that it is unsuitable for languages which are written from right to left such as Arabic or Hebrew.

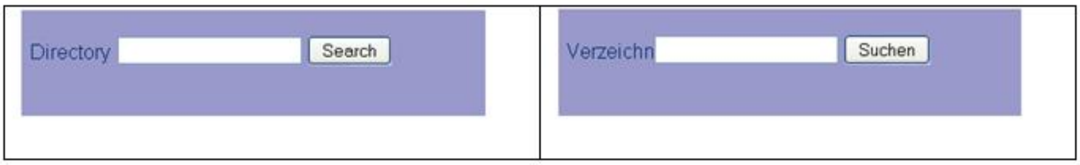


Figure 3: Interface layout which does not account for expansion

A better solution would be to arrange the screen of the original interface as follows since it provides plenty of free space to accommodate the longer German word.

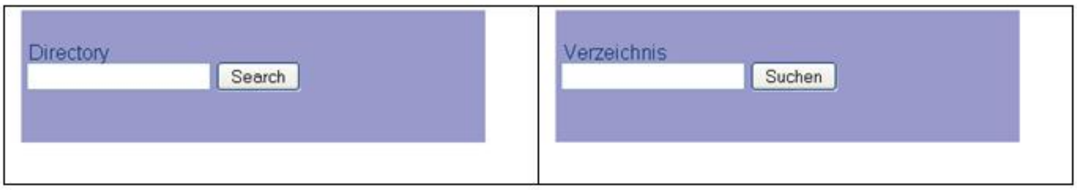


Figure 4: Interface layout which can accommodate text expansion

And of course, different languages mean that software will also need to use different dictionaries for spellcheckers and rules for checking grammar. Different languages, even among the fairly closely related European languages, use different conventions when writing numbers and currencies. Even among the group of countries which use the Euro as their currency, there is quite a lot of variation as regards where the € symbol appears in prices, with €17.20, 17.20€ and even 17€20 in use.

The issue of punctuation and numbers also affects the way in which dates, times and measurements are displayed in software or on websites. Where some countries use the comma "," as a decimal separator, others will use the decimal point ".". Although a seemingly small pointer, this could cause significant confusion for users, particularly when reading numbers. Indeed, the issue of dates is can also pose interesting challenges when localizing for countries such as Japan and Israel as well as various Muslim countries as they have traditional calendars which, although they run in parallel to the Gregorian calendar used in the West, are nonetheless in common use for important events such as public holidays etc.

When designing interface items asking users to submit contact information, developers need to consider not only the fact that different countries have different international access codes and area codes, but that telephone numbers may be of different lengths. This means that if a website is designed to check the validity of telephone numbers by comparing the number of digits against some predefined "ideal" or typical number of digits, anyone entering a number which is longer or shorter will find their number rejected even if it is a valid phone number. This also applies to post codes which may have differing combinations of numbers and digits as well as states and regions. Interfaces, therefore, need to provide sufficient flexibility to allow users to enter information in a variety of formats appropriate to their own region otherwise we might see, for example, a situation where a user from Denmark ordering from a US website is forced to choose from a list of US states when entering their address. This is not only a pointless exercise but may even lead to products being sent to the wrong address or the wrong sales taxes being applied to an order.

What is localized?

Up until the mid-1990s localization typically involved translating software user interfaces (UI), help systems and documentation. Nowadays it is much more extensive, particularly as a result of the World Wide Web which De Palma (2002) describes as the "8th continent" with over a billion inhabitants. Modern localization projects typically involve:

- **Software:** The translatable content in a typical software project will include text contained in the program code, manuals, leaflets, read me files, packaging, help systems, templates, installers etc.
- **Games:** Like conventional software projects, games projects include text contained in the game code as well as manuals, packaging and voiceovers.
- **Websites:** There are two key types of text to be translated on a website: "normal" text which is visible to users in a web browser and "hidden" text in the form of meta tags in the HTML code which are used by search engines, scripts, alt text for images and default text which is used to pre-populated forms. Other sources of text include databases which provide dynamic content for blogs, catalogues, e-commerce sites etc. and multimedia files such as Flash animations.

Depending on the product in question, the company responsible for it, deadlines and target markets etc. the scope and nature of a localization project can vary quite significantly. In the case of software the project will involve much more than simply translating the text strings displayed in the software's interface. As well as the interface text, software usually comes with a help system which needs to be translated as well as one or more user guides. Even the box the software is packaged in together with leaflets and quick start - collectively known as "collateral" - need to be translated and tested. In the case of documentation the finished translations will need to be sent for linguistic and DTP quality testing.

However, it's not unheard of for a product to be only partially localized. Helen Chandler (2005:12-14) in her discussion of computer games describes three different localization scenarios, which can be applied to other forms of localization:

Packaging and Manual Localization	This involves translating only the collateral material and documents but not the product itself. The main benefit of this approach is that it is relatively low cost and does not involve altering the software code. However, because the product itself is untranslated, users may not get the full benefit.
Partial Localization	In the case of games this involves translating only the text but none of the voiceover files. While it is more expensive and involved than packaging and manual localization, this approach is much mor

	accessible to the end user. The fact that voiceovers are not translated means that the costs associated with sourcing actors, recording dialogue, synchronising audio and video etc. are eliminated.
Full Localization	In this scenario everything is translated from collateral and interface text to graphics and audio. While considerably more expensive and time-consuming this approach ensures the best possible experience for end users.

Localization issues

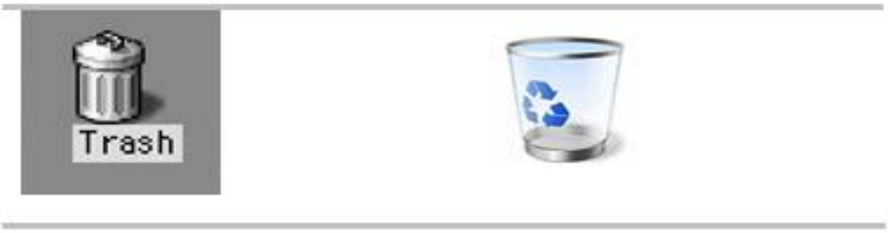
Once all of the preparatory work has been done during the internationalization stage in order to prevent some of the more fundamental problems and difficulties it's time to start localizing the project. One of the central tasks involved in this stage is the translation of textual components of the product but it also involves making various other modifications to the text, its presentation and various aspects of the interface and documentation.

Colours

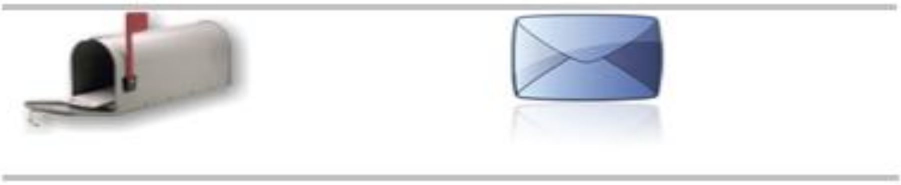
Certain colours have particular associations in different cultures and may need to be changed. For example the colour red in many western cultures signifies danger or stop whereas in Asian countries such as China it signifies prosperity and good fortune. Similarly, white is often associated with peace, life and purity in most cultures but in Japan it is associated with death and mourning black. Purple is often used to indicate royalty in Europe and the Middle East while in Latin America it signifies death. Hoft (1995:267-268) provides an interesting list of colours and their typical connotations in different cultures. The result of this is that, depending on the target culture and its customs, it may be necessary to change certain aspects of the product such as the interface colour, the colour of fonts and images in documentation and so on.

Graphics and icons

The issue of icons and symbols is something which can pose problems for localizers. Since icons are supposed to be immediately recognisable for users, it is common for the icons to be based on real-world artefacts with which the users are familiar. So far so good until it comes time to export the product to another culture which may or may not have the same real-world artefacts. In the early days of personal computers, users outside of the US often found themselves confused by the strange "Trash Can" icon on their desktop. The Trash Can was an early equivalent of today's "Recycle Bin" but unfortunately it didn't look like the types of bins non-Americans were used to using. Added to this the fact that "Trash Can" was a term not widely understood outside the US meant that the Trash Can was eventually given a makeover and so we have the modern Recycle Bin.



A similar problem arose in early email programs because of the decision to represent the inbox with a stereotypical American-style mailbox of the type typically seen perched atop a post at the end of someone's driveway. Again, not an image which is readily recognisable to non-Americans and this too was eventually replaced, often with a more recognisable envelope-style icon.



Issues of familiarity aside, icons can also cause offence if used without due regard for the cultural norms of the target audience. One seemingly harmless example is the "thumbs-up" shown in the image below. In many countries recognise this as a sign of approval or confirmation, and a graphic designer looking for a graphical way to let users know that they have successfully saved a file might be tempted to base an icon on this image. However, in modern day Afghanistan, Iraq and parts of Greece, Italy and France this simple gesture can be considered to be very impolite. In fact it is often regarded as the equivalent of the "middle finger salute" which is often seen in UK and USA. If such a graphic was to be used it would need to be replaced with a more culturally appropriate image in the localized version, for example a "tick mark".

As a general rule of thumb, localizers need to exercise caution when dealing with icons, particularly those which depict parts of the human body such as feet, faces or fingers, or even animals. Although a correctly internationalized product should not have any potentially inappropriate images, things sometimes slip through the net.

Care needs to be taken when selecting graphics or photos for illustrative purposes in documentation and on websites. In certain cultures, showing certain body parts such as the soles of the feet or the palm of the hand are considered offensive. It is also important that photos accurately reflect the ethnicity of the target audience. For example, a photograph intended to be used in the Japanese version of a website might not elicit the required response if it does not actually contain any Japanese people. Of course, much will depend on the level of multiculturalism of the target country but ideally photos should be as inclusive and representative of the ethnic mix as possible.

Multimedia Content

Software, websites and games often contain some aspects of multimedia content which extends beyond straight-forward text. Content such as voiceovers, videos and their captions, and PowerPoint presentations all require careful analysis and adaptation in order to meet the target audiences' expectations. One rather notorious example of what happens when this is not done involves a computer game produced by Microsoft which caused great offence to the Saudi Arabia government. The game's soundtrack contained chanting in the background presumably to add an exotic atmosphere to the game. It subsequently emerged that the chanting consisted of passages of the Koran which is strictly forbidden according to Islam. Microsoft later issued a new version of the game without the chanting, while keeping the previous versions in circulation because US staff thought the slip wouldn't be spotted, but the Saudi government banned the game anyway and demanded an official apology. Microsoft ultimately withdrew the game.

Textual conventions

In addition to the typical translation problems posed by technical texts, localization also presents some interesting challenges of its own as a result of both the technology being described by the text and by the technology used to do the actual translation. As we mentioned previously, most software applications use string tables which replace variables contained in sentences or phrases with predefined strings of text. For the translator, this can pose problems because the translated sentence has to make sense grammatically regardless of what text is added later when the variable is replaced.

Example: "Are you sure you want to delete the file $FILENAME?"

This is a relatively straight forward sentence to deal with. Problems can arise when more than one variable is used in a sentence or in certain languages when plurals are used. Ideally, of course, sentences should have been constructed in the source language in such a way that they minimise potential problems but this is not always done and the translator then needs to wrestle with the constraints presented by the variables and the grammatical and stylistic rules of the target language. Similarly, the fact that these strings are often translated in isolation using translation tools means that translators often do not have sufficient context in order to translate the text easily. Translators also need to ensure consistency of terminology between the documentation and the actual software because users will be confused if the documentation refers to a "Configuration" menu while the software has a "Settings" menu.

Culture

While much of what we have looked at so far can be attributed in one way or another to culture, there are very specific manifestations of the nature of an audience's cultural identity which need to be addressed in localization. Culture can be regarded as the sum total of shared knowledge, experiences, values, beliefs and aspirations of a group and its identity. This shared repository of knowledge gives rise to certain practices, customs and expectations which need to be incorporated, or at least catered for, in a localized product.

This might include, for example, conforming to particular forms of address, e.g. formal versus informal, polite versus familiar, etc. or cultural references and taboos. Issues such as the typical power distance (see Hofstede 1991) in a culture as well as an audience's propensity towards uncertainty avoidance may mean that a particular audience may not be accustomed to problem-solving in unfamiliar scenarios and may require additional assistance, whether in the form of more detailed instructions in documentation or a different screen design or layout which highlights, for example, the help section or contact details for technical support.

To sum up

Localization is a complex blend of textual, visual and intercultural communication which is both initiated and mediated by technology. The previous paragraphs present a very brief overview of various factors and issues involved in making technology available to multiple audiences. However, the most important things to remember are that localization is more than just translation. It is a complex, multidisciplinary process which requires a profound understanding of culture and language and how they affect people's perceptions of and interactions with technology. Localization is also one part of a much larger process which careful planning and a meticulous attention to detail. At its very core, localization is about recognising and accommodating the differences between different cultures to ensure that communication takes place as smoothly and effectively for both user and manufacturer.

Author

Dr. Jody Byrne is Lecturer in Translation Studies and Localization at the University of Sheffield where he teaches translation, localization, translation technologies and subtitling. He has published widely on various aspects of translation and technical communication and is the author of "Technical Translation: Usability Strategies for Translating Technical Documentation" (Springer,

2006). He is a practising technical translator and is a professional member of the Irish Translator's & Interpreters' Association and a Fellow of the Institute of Scientific & Technical Communicators.

References:

Best, Jo (2004) Causing offence costs Microsoft millions. [online] ZDNET. Available from: http://news.zdnet.co.uk/software/windows/0,39020396,39164112,00.htm [Accessed: 01/05/05]

Chandler, Heather (2005) The Game Localization Handbook. Hingham, MASS.: Charles River Media

De Palma, D. (2002) Business without Borders. New York: Wiley

Dunne, Keiran [ed.] (2006) Perspectives on Localization. Amsterdam: John Benjamins

Esselink, Bert (1998) A Practical Guide to Software Localization. Amsterdam: John Benjamins

Esselink, Bert (2003) A Practical Guide to Localization. Amsterdam: John Benjamins

Folaron, Debbie (2006) "A Discipline Coming of Age in the Digital Age". In: Dunne, Keiran (ed.) Perspectives on Localization. Amsterdam: John Benjamins, pp.195-219

Heimburg, Eric (2006) Localizing MMORPGs. In: Dunne, Keiran (ed.) Perspectives on Localization. Amsterdam: John Benjamins, pp. 135-151

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

Hoft, Nancy (1995) International Technical Communication. New York: John Wiley & Sons.

LISA (2007) The Globalization Industry Primer. [online] Available from: http://tinyurl.com/cnwxau.

LISA (2008) Localization [online] Available from: http://www.lisa.org/Localization.61.0.html [Accessed: 27 March, 2009]