Software Localisation: Some Issues and Challenges

M Sasikumar and Jayprasad J Hegde
CDAC, Mumbai (formerly NCST)
{sasi, jjhegde}@ncst.ernet.in

Abstract

As communication technologies are shortening geographical distances, businesses are aiming for global market penetration. At the same time, the benefits of Information and Communication Technologies are not reaching the lower rungs of the society. One major reason is the inability to use the software and systems currently available thanks to linguistic and cultural mismatch. Both these have led to tremendous impetus to localisation efforts aiming to alleviate these mismatches in use of software systems for various communities the world over. In this paper, we briefly outline the major aspects of this process, discuss some of the efforts being done in India, and think aloud on some of the open issues to be addressed. Of particular concern are the relatively little effort being done in the area of cultural localisation and the lack of empirical studies on the effectiveness of existing approaches for localisation.

Introduction

Localisation refers to adapting a software system (including websites) to a particular locale, so that it presents the image of a locally developed system aimed at that locale. As communication technologies are bringing geographically distant countries closer together, businesses are aiming at global market penetration of their products. This requires their products to be localised to the myriad communities of users across the globe.

Given that not even 10% of Indian population can communicate or understand English, the impact of localising major systems into various Indian languages cannot be overstated. The situation is no different in most developing and underdeveloped nations. This also means that if the benefits of Information and Communication Technologies have to reach below the top-layers of our society, we must adapt these technologies to fit into the 'world-view' of people in those sections of the society. The communities differ not only in terms of using different languages but also in a variety of cultural factors, such as the use of colours, meaning of gestures, symbols, analogies, etc. The major task, however, continues to be speaking the user's language. This involves primarily presenting menus, instructions, error messages and the entire product documentation including online help in all the languages relevant to the communities being addressed.

The alternative to localisation would be to teach all users to use a given system. There are two important reasons why this cannot be done: (1) the enormous cost involved, and (2) the gradual decline of the local culture if such a route is adopted. Training users from every different region is a very costly prospect, considering the range of languages, geographical distribution, and cultural variations. Note that apart from training on the use of the system, this will need training for adequate familiarity in the human language on which the system is based. One can safely discount many small organisations and startup companies from undertaking such a mammoth effort, assuming that mega-corporations could perhaps do.

The translation of source language content into different target languages is an effort intensive process. Ensuring a choice of target language expression for, say, a given menu list, requires not only expertise in both the languages but also a good understanding of the system being localised and familiarity with the culture and customs of the target audience. Machine translation technologies can and must play a significant role here. In addition, availability of language fonts of good quality and variety is also important to ensure a product localisation without major differences in look and feel.

Categories of software localisation

Broadly localisation efforts can be categorized as follows:

1. Capability to display/render text in the local language (or Display localisation): This aspect of localisation deals with the building of modules, capable of displaying text local in the script instead of the pervasive Roman Script. Most Indian scripts use a notation where the vowels when attached to a consonant, modify the consonant in the display form in a nonlinear fashion, rather than being displayed as a separate vowel. This introduces a plethora of complexities since keystrokes will not individually map to visible letters on the screen. Therefore, an additional keystroke may result in modification of something already on the screen or introduction of a new symbol.

2. Adapting the user interface to the local language (or Language localisation). This entails translating all the text material used in the system, including those in the documentation, to the local language, and modifying the software to use these instead of material in the original language. If the software is properly internationalised, the modification required may not be high. For example, if a one-word menu item translates to a long phrase in Hindi, it would need changes in the Screen layout and associated user interface.

3. Making the software culturally acceptable and convenient to the target community (or Cultural localisation). Cultural localisation is concerned about use of icons, metaphors, message conventions, etc. In rural areas, there may not be a notion of a *trash can*, and the trash can icon could be confused for a mail box, or a file folder. Similarly, the concept of a Desktop is itself considered a part of urban culture.

4. Development and use of input/output devices suited to the local languages and cultures (Device Localisation). Traditionally the most popular input device has been the QWERTY keyboard. This is quite unacceptable for entering phonetic rich, and alphabet rich languages such as those in India. Various mapping schemes have been invented for entering Indian language text using the existing keyboards. Of late, there has been a lot of good work in designing keyboard equivalents for entering Indian language text. This area has also attracted attention in the case of small-screen devices such as mobile phones, where even a QWERTY keyboard cannot be accommodated.

Much of the localisation efforts focus on display and language localisation.

Indian Efforts: A biased glance

A number of initiatives are under way the world over to address localisation issues effectively. There are collaborative ventures, research outfits (e.g., LRC) and associations (e.g., LISA) in addition to a number of companies offering turnkey localisation solutions. The size of localisation (including associated areas such as internationalisation) industry is estimated to be about 5 billion USD worldwide. This figure may increase substantially if the RoI on localisation investment – often reported to be around 10 to 1 - is sustained and adequate frameworks are established.

India is a culturally rich country and speaks many languages of which 18 are official. These 18 languages are written using 10 different scripts. There have been a lot of scattered attempts, particularly in India, to localise software to various Indian languages. Given the large number of languages in use in India, and substantial geographical spread of the country, this is perhaps, inevitable. Much of the localisation work seems to focus only on Linux and the core software components that make up the minimal desktop environment.

The Indian efforts in localisation have come from the academic and research organisations, as well as the industry. The academic centres along with the Government-funded R&D labs have contributed a lot to Display localisation. This contribution has come in the form of fonts,

encoding schemes, and support modules to display Indian Language fonts.

The earlier chaotic scene as far as encoding schemes has reduced with the entrance of the UNICODE encoding scheme. This is now the popular means for encoding Indian Language text.

To display Indian Language scripts, several implementations of "support modules" have blossomed all over India. Broadly there are two philosophies: whether the support for Indian Language fonts to be provided at the Operating System level, or whether it is to be provided at the application level.

The first approach -- that is, support at the OS level – deals with building the necessary components required for the Operating System to render Indian Language fonts. This could be in the form of building a renderer right into the Operating System. This approach is characteristic of NCST (now CDAC Mumbai) and the IIT Madras implementations. Microsoft and NCST collaborated to build the renderer for Indian Language fonts into Windows 2000. NCST has modified the X Windowing system (version 4.0.3) to render Open Type Indian Language Fonts. The IIT Madras "IndLinux" system which not only has a variant of X, as in the NCST system, to display Indian Language Fonts, but also has the kernel modified to display Indian Language Fonts.

The second approach -- that is, support at the application level -- deals with the usage of libraries which need to be present to display Indian Language fonts. The "pango" project, which provides a Open Source framework for the layout and rendering of internationalised text, and initiatives from companies like Sun and IBM for internationalisation in Java -- that is, providing functionality for rendering local language fonts in Java – are examples of this category.

NCST work and our approach

NCST has been a pioneer and an active player in various aspects of Indian language computing for many years. The work has dealt with Indian Language display, scripts for Indian language and the encoding scheme. A number of tools and technologies have come out of this work including tools for multilingual desktop publishing, designing fonts (Vinyas and Vidura), and multilingual text processing (Aalekh and Vividha).

Currently there is a focussed effort into the creation of support at the OS level for Indian Language fonts through projects like IndiX. This project not only involves the creation of renderers, but also involves the creation of fonts for more than ten languages. The Indix project currently has enabled Linux and an associated set of tools for six languages -- Hindi, Kannada, Malayalam, Marathi, Sanskrit and Tamil. A few more languages are in the process of being enabled. The approach has been to

enable the X-window system – the dominant user interface system for Linux – with Indian language support, which will directly enable all internationalized applications built on top of this to benefit from this capability. Unicode with UTF-8 is used as the base representation.

However, there are efforts at the application level too. Exemplars of such efforts are projects like BharateeyaOO.o a version of Open Office localised for Indian languages and available for both Windows and Linux; and Vartalaap, which is a multilingual communication software like Microsoft's Netmeeting.

BharateeyaOO makes use of the internationalisation framework provided by OpenOffice, the use of the highly customisable OpenType fonts, NCST's experience in shaping Indic texts, and adjustments in aspects like, for example 'locale', for creating an office suite specially meant for Indian language users. Vartalaap also makes use of similar technologies to bring the world of text messaging to the Indian language user.

Apart from projects related to display of Indian languages, transliteration and translation have also been significant area. Translation is concerned with human aided translation of sentences from English to Hindi (extendable to other Indian languages). Transliteration is required for correctly displaying non-regular words such as names of people, places, etc. Moving between the non-phonetic English and the phonetic Hindi, this is a challenging problem of high practical significance. Though, localisation was not the focus of these work initially, both these are important enablers for localisation efforts to be undertaken at a larger scale.

The transliteration project, called "Rupanthar", has an immensely practical application already: it is being used for the creation of bilingual degree certificates for the University of Mumbai. The Machine Translation project, called "MaTra", is being used as a subsystem for the creation of a Cross Lingual Information Retrieval System. Such a system would allow local users to access a document retrieved using a search engine like "Google" in his language.

Language resources in India

It is now widely accepted that fully automatic high quality translation system is not a realistic target. Different approaches have relied on varying levels of human intervention (human aided machine translation, as it is now called). The quality of translation produced automatically improves substantially if the domain is restricted (eg. Finance, healthcare, etc), or the language is restricted (simple sentences only, for example). One of the first successful MT systems was TAUM-Meteo, which translated weather forecast from English to French. One of the earliest and successful examples for using restrictive language is that of Systran for translating the manuals of Xerox Corporation. If one wants to

achieve translations for a general domain one has to provide for interaction with a human supervisor. The human supervisor would help in resolving ambiguities at various levels and indicate where a phrase should be attached.

The MT systems in India have been built keeping this in mind. Machine translation has been an active area of research in India from the 80s. There are many MT efforts on; some prominent ones among them are the projects at IIT Kanpur, University of Hyderabad, CDAC - Pune and Mumbai. The Technology Department in Indian Languages (TDIL), an initiative of the Department of IT, Ministry of Communication and Information technology, Government of India, has played an instrumental role in the rapid growth of language technology and resource development in India.

Since the mid and late 1990's a few more projects have been initiated - at IIT Bombay, IIIT Hyderabad, AU-KBC Centre Chennai and Jadavpur University Kolkata. There are also a couple of efforts from the private sector - from Super Infosoft Pvt. Ltd. and more recently, IBM Research Lab.

A lot of work has been done, yet the systems have not yet reached their full potential. The main obstacle has been the acute scarcity of basic lexical resources such as corpora, MRDs, lexicons, thesauri and terminology banks. Availability of such resources will also be of significant importance for localisation work in general. Also, the various MT groups have used different formalisms best suite to their specific applications, and hence there has been little sharing of resources among them.

These issues are being addressed now. There are governmental as well as voluntary efforts under way to develop common lexical resources, and to create forums for consolidating and coordinating NLP and MT efforts.

Issues and Challenges

In this section, we look at some of the issues to be looked into to make localisation efforts more goal-oriented and effective. These issues range from concerns of effectiveness of the models in use currently to building up specialized resources.

Is Language localisation really making a difference?

As mentioned earlier, there are a lot of scattered efforts within India itself to localise Linux and associated tools to various Indian languages. There is little interaction among these groups, as of now. One concern with such scattered work is the lack of sharing of know-how – including failures, pitfalls and successes – across these groups. This may lead to a number of mutually incompatible versions of the Linux environment. There are also multiple groups working on supporting a given set of

languages, sometimes following different philosophies – for example, the OS level changes versus the application level changes. It is important to work towards mutual compatibility of these approaches so that further work done on the resulting platforms do not lead to unnecessary branching effects.

Another concern with the localisation efforts in progress is the base for choice of words and phrases in the different languages. We are not sure to what extent this has been addressed by these approaches. Consider the focus on localisation primarily to reach the Information Technology benefits to the non-English speaking community. Note that even in English, the menu terms, commands, etc. used do not necessarily have strong connection with the meaning of the term in the English language. This necessitates the need for training to alleviate the kind of difficulties that beginning programmers face when encountering expressions such as 'x = x + 1'.

Unless proper care and effort is not exercised in choice of equivalent phrases in local languages, we may have a much worse learning curve, negatively affecting our efforts. We have noticed that often the attempt is to choose words based on the current English word, and this often leads to very unintuitive images for the intended user who is unfamiliar with English. For example, the word 'administer' when translated to Malayalam would generally get a meaning as in the usage 'Indian administrative service' (that is, something like 'govern'). The notion of someone *governing* your computer system will make no sense to a Malayalam speaker! One needs to look deeper to find out the intended interpretation of 'system administration' and then base the translation on that. As another example, the word 'save' is generally used to 'store in a permanent form'. But 'save' generally has no such sense in its meaning as a normal English word. So this incompatibility gets propagated and perhaps multiplied when moving to other languages.

At one extreme this borders on issues of cultural localisation itself. When choosing target language expressions for a given entity/action, ideally we need to find metaphors or scenarios commonly associated with the intended entity or action, in the local community.

Is cultural localisation required/meaningful?

While there have been a lot of talk about cultural localisation in the literature, little seems to have been implemented. One level of cultural issues is today being taken care of by locale setting in many of the software systems. An example of this would be the date display format (mm-dd-yy or dd-mm-yy), which can be configured once on your machine and all properly internationalised applications will see the change without need for individual configurations. Similarly, left-handed user settings can also be done once.

The next level of cultural issues includes significance of various colors in various cultures, interpretation of facial and hand gestures (for example, as used in chat systems, emoticons, etc.), variation in meaning of words, etc. Further, we need to look at ability of images and icons to signify the intended meaning in various regions and to various user groups. These may vary not just by the geographical areas, but also depending on your religion.

Supporting such kind of localisation vary in complexity from the trivial (look up a colour mapping table, for example) to the nearly impossible. For example, it is difficult for product names to be changed to suit different regions. Beyond citing examples of possible mismatches that arise, we need documented analysis of these issues and their implications. At some level, it may be better to address some of these incompatibilities by suitable documentation or training programmes. What issues significantly impact usability and what are 'worth having'? Systematic studies of the various types of cultural mismatches spanning multiple cultures would be of importance in answering these, so that one can focus the available energy to the important problems.

As hinted earlier, we need to document experience available in this regard from scattered projects, with a view to share them across the world. This will help to evolve general enough approaches for addressing these issues. Note that localisation problems are addressed in a two fold approach: by producing guidelines for internationalising software thereby minimizing core changes in the software, and applying localisation techniques possibly involving code changes. Moving whatever is possible to the internationalization stage helps to reduce the localisation overheads, which is proportional to the number of localised versions.

CDAC Mumbai is initiating a project to build up such a knowledge base. Inputs regarding the scope, and structure, as well as actual content are welcome.

Natural Language Processing for Localisation

NL techniques, particularly, translation plays a major role in localisation efforts, particularly in Language localisation. Automated assistance in this area would be valuable.

As mentioned earlier, there has been a lot of work in the area of machine translation the world over during the last few decades. Good working models are available for some of the language combinations. Languages that differ widely in structure are still open areas of research. Given the current state of open-ended translation systems, most localisation vendors are currently using translation memory systems and support frameworks for carrying out the translation work.

Localisation brings in interesting challenges particularly in the field of machine aided translation. One-to-many translation as opposed to

works assuming a language pair model is of immediate interest. Effective use of context will be a necessity to obtain useful results. Another major challenge is translating short crisp phrases and the need to obtain compact translation without losing clarity.

Conclusion

Thinking aloud on the issues raised here, a number of further lines of action emerge.

1. We need effectiveness studies for various target groups for the existing localisation efforts. This will help us to know the real impact and potential of localisation, and for feedback on our lines of thinking.

2. Comparative studies of various localisation efforts are also needed at the technical level. This will help partly to jump-start new initiatives more effectively, and partly to evolve the field into maturity.

3. Large-scale cooperative ventures are required to build up knowledge bases for localisation – across languages and cultures. These include linguistic resources (which are already in progress at many places) as well as cultural knowledge bases. This is an area where individualised efforts are unlikely to pay off well, since our focus is movement of content from one culture to another.

References and Useful Links

The localisation industry primer (2nd edition), LISA, http://www.lisa.org/

Internationalisation and localisation of software, George Calzat, 1996, citeseer.nj.nec.com/calzat96internationalization.html

Localising web sites and content on the internet. SDL International. http://www.sdlintl.com/

Mozilla Hindi Translation Project:
http://www.bttlindia.com/mozilla/contrib.html

The Pango project: http://www.pango.org

Display Localisation
http://marketing.openoffice.org/conference/presentations-pdf/IndianPerspective.pdf
http://trinetra.ncb.ernet.in/bharateeyaoo/
http://trinetra.ncb.ernet.in/vartalaap/
http://staff.ncst.ernet.in/shrinath/indicFeb1999.pdf

Cultural Localisation

http://toastytech.com/guis/bob2.html
http://www.wired.com/news/technology/0,1282,61265-
2,00.html?tw=wn_story_page_next1
http://www.people.virginia.edu/~jrd8e/MDST110_F03/homepage.html
http://www.e-culturefair.nl/site/index_en.html

metaphors:
http://www.cs.nott.ac.uk/~mzt/MetaphorAndHCI/
http://delivery.acm.org/10.1145/510000/505107/p7-
marcus.html?key1=505107&key2=4934601701&coll=portal&dl=ACM&C
FID=14803862&CFTOKEN=60971555