

FOSS Localization: A Solution for the ICT Dilemma of Developing Countries

Syed Waqar ul Qounain Jaffry
Punjab University College of Information
Technology, Lahore, Pakistan
swjaffry@pucit.edu.pk

Umer Riaz Kayani
Punjab University College of Information
Technology, Lahore, Pakistan
umerkyani@pucit.edu.pk

Abstract

Information and communication technology (ICT) has tremendously expanded over the last three decades making the access to right information at the right time feasible ensuring the success of an individual, organization or culture. In order to make the most out of this exciting revolution one must be in a position to afford and completely comprehend what is offered by this technology. Unfortunately most of the software are controlled by proprietary that are economically unaffordable for developing countries and are based on language that is not comprehensible by their masses. Software localization of Free and Open Source Software (FOSS) is an effort that addresses this twofold dilemma. FOSS made software affordable while localization bridges the language barrier that helps people to fully comprehend and utilize the benefits of ICT. In this research we have explored various aspects of the software localization of free and open source operating system (FOSOS) and developed a working prototype. Paper explains concept and all the technical steps of FOSS localization of Ubuntu Linux that is a FOSOS with a foreseeable future work.

1. Introduction

Today, the advancements are the result of technological breakthroughs happening throughout the world. If one has to choose an agent that has contributed most in human development, it would, without any doubt be Information and Communication Technology (ICT). ICT has revolutionized the way of living on the globe, so if a culture wants to cope with challenges of globalization and emerging technologies then she has to embrace whatever is new and advanced in it.

The ICT dilemma, developing nations are facing right now, is twofold in nature. Firstly the high costs of the proprietary software is unaffordable by the many facets of different cultures and secondly very small

community of developing countries can read and write English, thus leaving behind a major portion of population which understands their native languages but is unaware of the technological advancements because of the language barrier. Solution to these problems is a fully Localized Free and Open Source Software Platform that not only eliminates the cost factor but also bridges language barrier.

Urdu Linux Localization Project is a research project that aimed at people with understanding of Urdu to fully benefit from the ICT. This research focuses on Ubuntu Linux as a prototype which is a Free and Open Source Software. It is free to download and install and can possess a large user base on the Internet with technical support that leads to the solution of cost and language interface. This paper is divided into three sections that explain concept of FOSS, software localization process and Urdu Linux Localization Project (ULLP) that is the prototype as the solution for the ICT dilemma of developing countries, with a conclusion on the foreseeable future work.

2. Free and Open Source Software

Free and Open Source Software (FOSS) is a software for which the human-readable source code is made available to the user of the software without any cost, who can then modify the code in order to fit the software to the his/her needs. The source code is the set of written instructions that define a program in its original form, and when it's made fully accessible to use, Programmers can read it, modify it, and redistribute it, thereby improving and adapting the software [1]. In this manner the software evolves at a rate unmatched by traditional proprietary software.

Over the years free and open source software has been gaining momentum, that is now causing quite a stir in the commercial world as large software corporations are finding themselves competing against commercially available open source software. This

new demand for free and open source software has generated great interest among the scholars and researchers in disciplines ranging from sociology to economics to social psychology, and has raised questions in fields of application ranging from innovation processes to strategic management. This interest has helped in advocating the usage of FOSS in the both Public and Private sectors mostly benefiting the non-for-profit sectors in developing countries. FOSS offers solution to many problems of the developing world including high cost of software technology, software piracy, and monopoly of software giants that ceased the expansion of local software industry.

3. Localization

Software localization is a process of translating software user interfaces from one language to another and adapting it to suit a local culture [2].

Localization can occur after the software application has been successfully used in one culture and its developer wants to make the necessary changes for a new culture. Although this produces software that is tailored to that specific culture, the application is still not ready for easy adaptation to other cultures [3]. Increasingly, localization is a concern from the beginning of application development, with the business strategy pushing to introduce the application in multiple cultures. In this second approach, the application is designed from the beginning to support international conventions, languages, formats, and processing. This approach makes localization easier and less costly [4], even though it requires a larger initial cost [3].

The distinction between internationalization and localization is subtle but important. Internationalization is the process of designing an application so that it can be adapted to various languages and regions without engineering changes [5]. Internationalization occurs in the country where the product is originally developed. It is not uncommon for development groups to focus only on elements related to text, numbers and dates.

Localization refers to the process of infusing a specific cultural context into a previously internationalized product; for instance, translating English text and message files into Urdu for Urdu users. Like internationalization, localization is usually limited to translating the text, date and number formats [6]. But creating a product that speaks fluently in another culture involves more than this. Properly localized software applications, just like properly localized automobiles, toasters, beverages, and magazines, reflect the values, ethics, morals and

language (or languages) of the cultures under consideration [7].

Users can interact with a successfully localized product in their own language and in a setting that feels natural to them. This means, for instance, that all messages are in their own language, that they can input names, addresses, dates, and other data in the same way they would write them down on paper, that they can freely use their standard keyboard characters wherever an entry can be made, and that any error messages are comprehensible to them rather than representing English born-speakers. Localization comprises the complete translation of program interfaces and any online or printed documentation of it [8].

3.1. Software Internationalization

To develop software for different cultures worldwide, it should pass through internationalization and localization processes [9]. Internationalization is the process that separates the software into two components, a culture-independent and a culture-dependent component. The culture-independent component, known as the generic core, contains the bulk of the software and excludes culture-sensitive elements.

3.2. Software Localization

Software localization provides the culture-dependent component of a particular target culture. The culture-sensitive elements, comprising dialog messages, error messages, and menu names are localized (translated) and stored in a message file. There is a different message file for each culture. If the software is required in a new language, only the localization process takes place; there is no modification of the software's generic core (culture independent component of the software). Maintenance is also easier as only the generic core is modified. Thus, to realize software for the worldwide use, the global-software development lifecycle (GSDLC) [10] is proposed, which includes internationalization and localization processes.

3.3. Locale

To support users in different cultures, programs must not only use translated text, but must also be adapted to local conventions called locale. These conventions differ by language or region and include the formatting of numbers, dates, times, and currency values, as well

as support for differences in measurement units, text sorting order, and other data formats and services.

3.4 Software Localization Process

Software Localization Process (SLP) can not be explained without the mention of the Software Internationalization Process (SIP). When one says that a particular software is Localizable, it means that this software is already Internationalized, that semantically means the application/software can be easily ported/adopted to a particular language. The software/application has been generalized and there are standard procedures to adapt it to a particular language. The software can be localized in any language, of user's choice, by using these standard approaches or procedures which is called Software Localization Process [13].

In many cases locale design is not enough to localize software for a particular language. One has to start from scratch to build up whole computing architecture to support all computing facilities for a particular culture. Mostly a naive computing culture requires the following fundamentals to be computing enabled,

1. definition of language Code Page and its acceptance from UNICODE consortium
2. design of language font
3. design of language keyboard layout

Software localization of a computing enabled culture involves the following steps,

1. design of language locale
2. translation of software related text and its association with in system

4. Urdu Linux Localization Project

Urdu Linux Localization Project (ULLP) [14] is Free and Open Source Project that address the two main problems that are real hindrance for the adoption of information and communication technology in developing countries i.e. software affordability and its availability in local language. As software technology is rapidly becoming one of the most fundamental building blocks of human interaction and activity and becoming vital in national economic development, so to void these barriers is mandatory.

ULLP used the philosophy of FOSS and has started to work on localization of software platform that resides under GNU-GPL. Linux was selected for the purpose as it is available under GNU-GPL, widely used and gone through the software internalization process. Ubuntu [15] variant of Linux was used as first prototype platform as its Urdu user guide [16] was

available online for public use and it contains fairly simple localization process. In the future other variants of Linux will also be localized based on their popularity.

ULLP has successfully completed Urdu font design and installation, phonetic keyboard layout installation, system wide Urdu input/output support, semi translation of desktop and start menus (see Figure 1). Pipelined activities for the project includes design of different Urdu fonts, translation of messages, errors and applications dependant text, and formation of Urdu enabled Linux distribution.



Figure 1 - Urdu enabled Text Editor and Translated System Menus

5. Ubuntu Urdu Linux Localization Steps

Following steps are being followed to enable full Urdu support in Ubuntu-Linux operating system.

5.1. Installation of Urdu Keyboard

To install Urdu keyboard layout follow the following steps:

1. Download Urdu keyboard layout from the following site [17]:
www.fossfp.org/fossac/downloads/ullp/ur
2. Copy this file to following locations:
/etc/X11/xkb/symbols
/etc/X11/xkb/symbols/pc
3. Edit "xorg.lst" placed in
/etc/X11/xkb/rules
and add "ur Urdu Pakistan" line as shown below:

```
.  
gb United Kingdom
```

th_tis Thai (TIS-820.2538)

th_pat Thai (Pattachote)

tr Turkish

ua Ukrainian

ur Urdu Pakistan

4. Compile this new layout into X Server by executing following commands:

```
cd /etc/X11/xkb/symbols
xkbcomp -lhlpr '*' -o ../symbols.dir
```

5. After this follow the following menu
“Applications -> System Tools ->
Configuration Editor”

and add “ur” (without quotes) in the list of already installed layouts in the path:

```
/desktop/gnome/peripherals/keyboard/kbd
```

5.2. Installation of Keyboard Layout Switcher

By clicking on keyboard layout switcher, you can easily change between English or any other language layouts for keyboard input. You can install keyboard layout switcher by going to panel, right-clicking and selecting

Add to Panel->Keyboard Indicator

Once Urdu keyboard layout and layout switcher is installed, you can write Urdu anywhere in Linux including Firefox and Thunderbird by switching to Urdu layout.

5.3. Installation of Urdu Fonts

Only true type fonts are so far tested to be successfully integrated with Ubuntu X font Server. Just paste the fonts files *.ttf into the fonts folder, and then test the fonts installation in an editor. Also remember to select your desired font in text editor.

Download the following GNU-GLP Unicode font from the following link [18]:

<http://tabish.freeshell.org/u-font/>

5.4. Installation of System wide Language Support

Use the following command as root for installing Urdu language locale.

```
sudo dpkg-reconfigure locales
```

Check your required language locale and select ok. Re-login with the respective locale (use language tab at the login screen).

5.5. Translation

Once first four phases are completed you can now start the translation process that includes following steps.

1. visit the Debian website [19] and download the PO (portability object) files of desired application.
2. replace the original strings to be translated in the PO files with the Urdu string by using the Urdu keyboard.
3. convert the PO files to MO (machine object) files with the following command as root user:

```
msgfmt POFileName
```

4. now place the .mo file at the following location for system to recognize the translated messages.

```
/usr/share/locale/ur/LC_MESSAGES
folder, re-login to see the effects.
```

6. Conclusion

In last few decays ICT has proved its worth as most prominent agent of socio economic development. Developing countries are still lagging behind due to high cost of this technology and unavailability of software in their native language. In this research we have proposed the solution to this problem as fully localized free and open source software platform for computing, so that all nations can fully utilize the benefits of these emerging technologies. In our research we have designed a prototype for Urdu localization of Linux based platform under Urdu Linux Localization Project (ULLP). ULLP is a Free and Open Source Software Development Project. It is licensed under the GNU-GPL. The project aims at creating a Linux distribution that will fully support the Urdu language. Currently the project has enabled Urdu for computing in Ubuntu-Linux Operating System. Pipelined activities for the project includes design of different Urdu fonts, translation of messages, errors, applications dependant text, and formation of Urdu enabled Linux distribution. It is strongly believed that the solution proposed in this paper i.e. fully localized FOSS platform is the most viable solution for socio economic development of developing countries that voids their ICT dilemma. It is hoped that case study of ULLP will help other developing nations to make software comprehensible and affordable for their masses.

7. Acknowledgements

Authors extend their thanks to Free and Open Source Software Foundation of Pakistan (FOSSFP) and all those volunteers that are directly or indirectly contributing in Urdu Linux Localization Project (ULLP).

8. References

- [1] Anousak Souphavanh and Theppitak Karoonboonyanan, "Free/Open Source Software: Localization", Academic Press, Butterworth-Heinemann.
- [2] Wikipedia free encyclopedia, (10 August 2005) [Online], "Software Localization Definition", Available: http://en.wikipedia.org/wiki/Software_localization, 14 August 2005 [date accessed]
- [3] D. Johnston, "Culture Shock: Taking Your Software Products International," AS/400, May-June 1996, reprint.
- [4] L.C. Miller, "Transborder Tips and Traps," Byte, vol. 19, no. 6, June 1994, pp. 93-102.
- [5] Sun Microsystems, "Software Internationalization Page", Available: <http://java.sun.com/j2se/1.3/docs/guide/intl/>, 14 August 2005 [date accessed]
- [6] Patricia Russo, Stephen Boor, "How Fluent is Your Interface? Designing for International Users", ACM INTERCH'93.
- [7] Taylor, D. "Global Software", Springer-Verlag, New York, 1992.
- [8] Per N. Dohler, "Facets of Software Localization A Translator's View", Translation Journal 1997.
- [9] Madell T, Parsons C, and Abegg J. "Developing and Localizing International Software", Hewlett Packard Professional Books, 1992.
- [10] Alvin W. Yeo, "Global-Software Development Lifecycle: An Exploratory Study", SIGCHI'01, Seattle, WA, USA, March 31-April 4, 2001.
- [11] UNICODE Consortium, (21 June 2005) [Online], "Common Locale Data Repository (CLDR) Project", Available: <http://www.unicode.org/cldr/>, 12 August 2005 [date accessed]
- [12] Internet Assigned Number Authority, (15 July 2005) [Online], "Registration of Special RFC 3066 Language Tags", Available: <http://www.iana.org/assignments/language-tags>, 13 August 2005 [date accessed]
- [13] International Open Source Network Centre for Development of Advanced Computing, "How-to Guide for Localisation", Mumbai.
- [14] Free and Open Source Software Foundation of Pakistan, (10 July 2005) [Online], "Urdu Linux Localization Project", Available: <http://www.fossfp.org/ullp>, 21 August 2005 [date accessed]
- [15] Canonical Ltd, (20 July 2005) [Online], "Ubuntu - Linux for Human Beings", Available: <http://www.ubuntulinux.org/>, 15 August 2005 [date accessed]
- [16] Chua Wen Kiat, Hameed, (20 July 2005) [Online], "Ubuntu Urdu Guide Home Page", Available: <http://ur.ubuntuguide.org/>, 8 August 2005 [date accessed]
- [17] Mr. Nadir Durrani, (15 July 2005) [Online], "Urdu Keyboard Layout", Available: <http://www.fossfp.org/fossac/downloads/ullp/ur>, 10 August 2005 [date accessed]
- [18] Shehzad Ali and Tabish, (1 July 2002) [Online], "Unicode Nastalique Font under GNU-GPL", Available: <http://tabish.freeshell.org/u-font/>, 10 July 2005 [date accessed]
- [19] Debian Software in the Public Interest, Inc, (10 June 2005) [Online] "PO Files for Urdu Internationalization", Available: <http://www.debian.org/international/110n/po/ur>, 11 August 2005 [date accessed]