Localization of Open Source Web Content Management System

Rufael Tadesse
HiLCoE, Computer Science Programme, Ethiopia
Computer Revolution Africa, Ethiopia
rufael.tadesse@gmail.com

Fekade Getahun
HiLCoE, Ethiopia
Department of Computer Science, Addis Ababa University, Ethiopia
fekade.getahun@aau.edu.et

Abstract

Content Management Systems (CMSs) are being applied by several firms for the purpose of facilitating document control, auditing, editing and managing timeline. Most CMS provide Web-based publishing, format management, revision control (version control), indexing, search, and retrieval. The open source CMSs are becoming popular due to low cost, easy customization, easy to use, workflow management and built-in support for foreign languages. However, users using local language are unable to understand the system and several components related to it. Focusing on the localization aspect of CMS, several works have been done. The works vary on the CMS selected to be localized and the approach followed during the localization process.

In this paper, the objective is to localize CMS into Ethiopic languages and enable non-expert users to be site developers, content contributors and content creators. Various CMSs are assessed and Joomla Web Content Management System (WCMS) is selected to be localized. Templates and modules based localization is adopted and basic components of CMS such as entities, concepts, and ideas are localized into Ethiopic languages (Amharic, Oromigna and Tigrigna) by creating a language pack. To empower developers and allow users in writing using local language, a virtual keyboard for Amharic is also designed and embedded in to the CMS. The localized Joomla has been given to end users and programmers to design their own website. They developed the provided websites using the localized Joomla in ways required and made use of the designed virtual keyboard with relative ease.

Keywords: Open Source Software; Content Management System; Localization; Joomla Web Content Management System; Language Pack; Translation Component; Virtual Keyboard

1. Introduction

Recently, free and open source software have become common in many countries. Open-source software (OSS) is a software with its source code made available and licensed with free or affordable license fee in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. Open-source software is very often developed in a public, collaborative manner. Open-source software is the most prominent example of open-source development and often compared to (technically defined) user-generated content or open-content movements [1].

The concept of open source has also been applied in WCMS to facilitate the design, development, implementation, and publication of content to a website and avoid the need for the user to possess technical knowledge of editing hypertext mark-up language (HTML) files. Such systems allow content creators to directly add or update content on a website without involving an Information Technology (IT) department or external company. WCMS, therefore, empowers members of each department within an organization to update their own sections on a website thereby reducing the number of people involved in the end-to-end process and ensuring that timely updates can be made [2].

In Ethiopia, the number of people using mostly native language as a form of communication tends to cover most of the country's population. According to the 2007 Ethiopian census, most people speak

Oromigna, Amharic; Somaligna, Tigrigna; Sidamigna or Sidaamu Afoo, Wolaytta, Guragigna, Afar: 'Qafár af' and other languages [3]. Due to this, many people face challenges in using and adapting technological solutions due to the required knowledge on foreign languages. One of the solutions to inculcate such population to the use of technology is to localize systems into local context.

Localization is the process of adapting, translating and customizing a product (software) for a specific target [4]. Localizing Open Source WCMS is performed to empower users into creating, managing and editing content using local languages such as Amharic. WCMSs have different kinds of components embedded within a package

Considering a real world scenario, web content programmers prefer to work on open source CMS than proprietary WCMSs due to cost, environment restriction, ownership and flexibility in modifying core functionality as required. However, content developers still face problems in understanding and making good use of predefined components of the WCMS as local language is not handled. Most feel comfortable if they have readymade components in local language that can be embedded into a website rather than using components built in foreign language.

The best option to facilitate such requirements is to localize the WCMS including its components. Customizing Open Source WCMS together with its components to support local languages enables users (programmer or naïve user) with preference of using local language to operate and develop a web site easily.

The rest of this paper is organized as follows. Section two discusses related works. Section three presents the proposed solution in which this research aims to address and describes the main parts of the solution. Section four focuses on discussion on the findings and comparison is made in relation to similar works. Section five summarizes the paper and provides areas for future works to be focused on.

2. Related Work

Several works have been done in the area of localizing open source software. Each work differs from the other based on the set objective and the selected content management system. Related to local context, there are two prominent localization attempts. In [5], the authors presented an approach to localize Open CMS for managing documents electronically, improving the internal operations of an organization, raising community satisfaction, and simplifying document management and exchange. The project base its domain on only Ethiopian governmental organizations in the process of the Open CMS alteration and puts more focus on trying to use the CMS for document management.

In [6], localizing Joomla content management system 1.5 into Amharic to make professional and non-professional individuals to take part in web site development has been conducted. The front and back end interfaces are translated into Amharic and a virtual keyboard is designed to make Amharic text entry easier. The developed language pack and designed virtual keyboard is compatible to work with Joomla version 1.5 and cannot be imported into the latest version of Joomla. The author points out localizing extensions and templates as part of future work.

This research addresses localizing extensions and templates and lays down an approach to extend the extension to other Ethiopic language as well.

3. The Proposed Solution

The main objective of this research is to localize a CMS by identifying the popular CMS and assessing users' experience in using the CMS. In order to identify the popular CMS, ten CMS products were selected based on a study that uses WAP Analyser which unleashes WCMS technologies used. A total of eight criteria have been considered for evaluating the CMSs where seven of the criteria are documented by different researchers (i.e., Usability, Performance, Security, Built in applications, Flexibility and Ease of Customization Management and Interoperability and Documentation and Support) and one has been added to cover the requirements and the set objectives of the research (Modules/Plug-ins Extensibility and Template Usage). CMS matrix and Weight scoring method has been applied on the ten WCMSs to select the top three WCMS. After the top three had been selected, a questionnaire was distributed to thirty website development companies in order to gather data related to use, experience and popularity of CMSs in the Ethiopian context.

The proposed solution focuses on providing a localization means for the selected CMS, i.e., Joomla. One part of localization that this research covers is minimizing the gap between end users and translation values with in Joomla. The general architecture used to localize the CMS is shown in Figure 1. The architecture is a layer based architecture and comprises an interface, database, authentication/access checking and management for front-end, backend and package creation. The interface layer serves as a line between end users and the developed component whereas the authentication/access checking layer bases on a security feature provided by Joomla to authenticate an end user before using the component. Layers for the management provide the availability of Package creation, Front-end and Backend management after an end user has been authenticated. The translation database holds data translated by end users for local languages.

The translation component is a tool that is developed to support a translation mechanism for the newer version of Joomla to support Amharic, Oromigna and Tigrigna languages. The architecture for the translation component is in alignment with the architecture of Joomla where the emerged component is embedded on the external layer of Joomla architecture. The component is designed to fill the gap that exists within the language strings and elements within Joomla architecture. The translation component has package creation and language management subsystems. The access control and security is dependent on the security structure for backend administration of Joomla. Layered architecture is used for the interfacing of subsystems and organises the component into a set of layers. The component is developed in a way to convert file based structure into a database structure by storing all language related strings in a database.

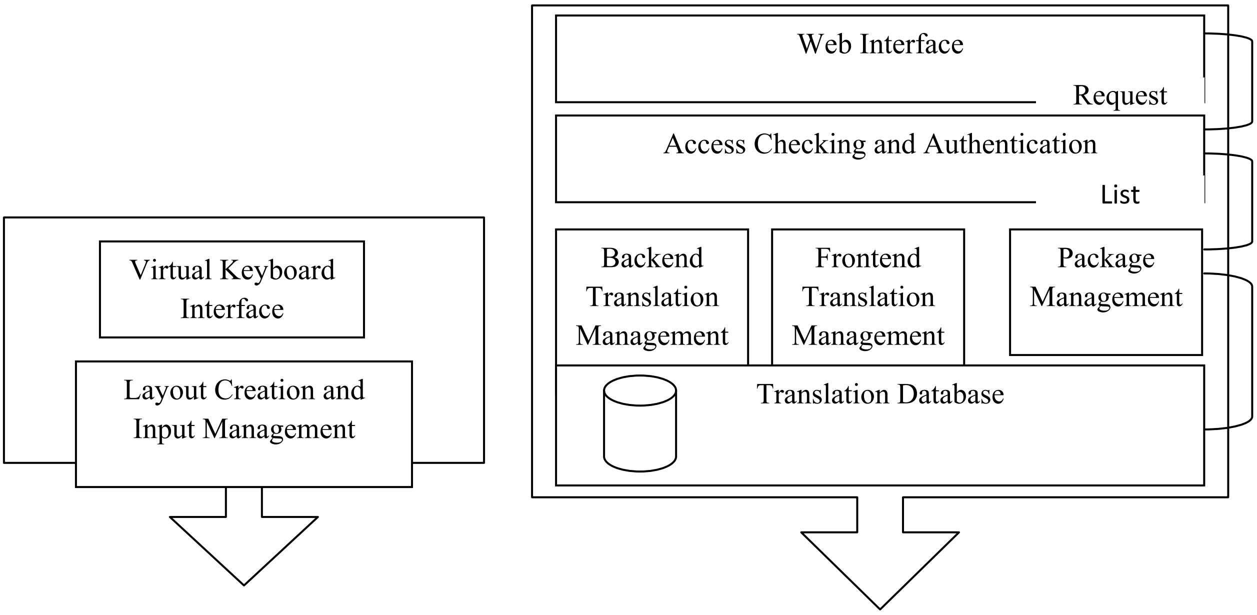
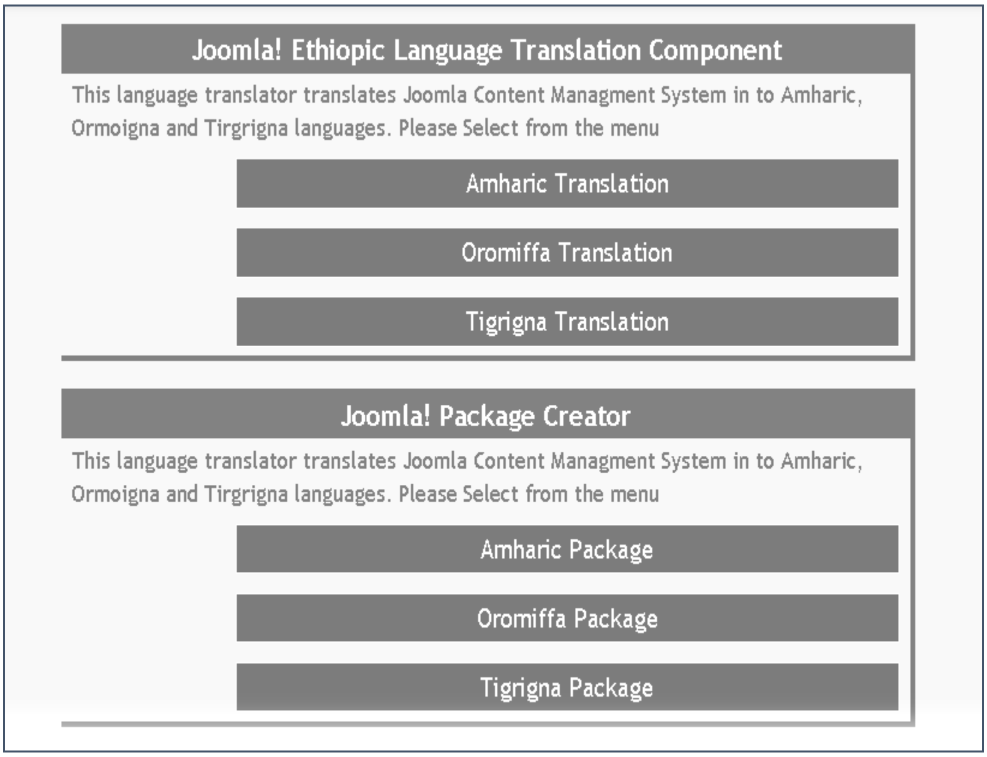

Web Interface
Request
Access Checking and Authentication
List
Backend Translation Management
Frontend Translation Management
Package Management
Translation Database
Virtual Keyboard Interface
Layout Creation and Input Management

Figure 1: General Architecture

A translation database is designed to match the file based language files and categorized in a way to enable users access strings found with in a file on a well-constructed and easy to manage interface. The database works in collaboration with the component developed. The component contains an interface that enables users to translate strings into the language that they need. This capability is shown in Figure 2. The component reads the default value from the database and provides a space for localized values to be entered. The component is developed in a way to be easily installed and used in a friendly manner. The component allows translation for three Ethiopic languages. It enables users to translate, edit and create language packs for Joomla. The created language packs can then be imported into Joomla and be used. The translated values are stored based on the table structure that is created for each interface.



Joomla! Ethiopic Language Translation Component

This language translator translates Joomla Content Managment System in to Amharic, Ormoigna and Tirgrigna languages. Please Select from the menu

Joomla! Package Creator

This language translator translates Joomla Content Managment System in to Amharic, Ormoigna and Tirgrigna languages. Please Select from the menu

Figure 2: Translation Component Main Menu Interface

This component allows a user to create a language pack for a specific language selected. Each pack includes *ini* files and localize files. The language package will be provided as an input to the content management system and installed to apply a selected language. A translation pack consists of an xml file, a blank index file and a set of *.ini files containing the language strings for a language. The easiest way to understand this structure is to look at the files in the language/en-GB folder of Joomla. The zip files contain xml files and set of ini files. Joomla language INI files must be saved as UTF-8. The format of language definition files is very basic. Blanks lines and lines beginning with "#" are ignored and the latter may be used to add comments to the file. Each line consists of a pair of strings separated by an equals sign denoting key value pair represented in the following pattern:

KEY=”Value”

where “KEY” is a string to be translated and “Value” is the translated string. For example if set of lines can be taken from the ini files,

COM_BANNER=”Banner”

ADDITIONAL_INFORMATION=”Additional Information”

The values contain strings that are displayed in Joomla. The values can be changed to Unicode values and the modified values will be displayed in Joomla.

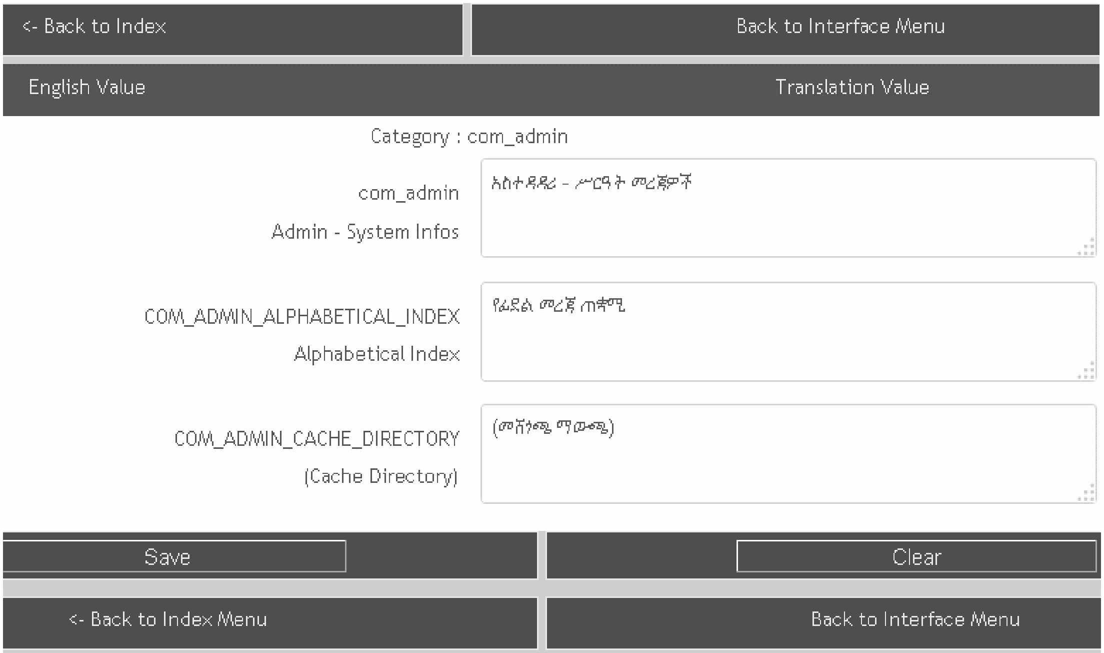
COM_BANNER="ሰንደቅ"

ADDITIONAL_INFORMATION=”ተጨማሪ መረጃ”

When Joomla reads the set of ini files, The “KEY” that is the first word before the equal sign on every line must be in all capitals or Joomla will not be able to read the string for that particular item. The

"KEY" can include spaces and other punctuation characters but there should not be any spaces in either side of the equals sign as spaces are significant. If more than one entry has the same left-hand side, the last one to be encountered is the one that will be used. The "VALUE" cannot include double-quote characters ("). To get a double-quote character, use the HTML special character sequence "&quot;" instead. Single-quote characters (') are valid [7].

Localization for Joomla is maintained through basic steps to change the working environment into a specified language selected by the end-user. Joomla uses three language packages, one for the frontend, one for the administrator and one for installation. Ideally, all three must be created for a full package. A full language pack will contain translations for site and administration parts. The translation files contain translation for components, plug-ins, modules and templates where each of these files is attached with the interface and every functionality of Joomla. The files are packed in to a zip file and can be provided to Joomla as a language pack. Once the language pack has been imported, a user can change the language that is required to be used for the overall operation just by setting the language to Amharic. The translation of the every term is based on an ICT glossary for Amharic, Oromigna and Tigrigna which was done by ICT Development Office and Ethiopian Languages Research Center (ELRC) [8]. Figure 3, shows the translation form that allows responsible persons to maintain the value of administrative units.



Category : com_admin

English Value	Translation Value
com_admin Admin - System Infos	አስተዳዳሪ - ሥርዓት መረጃዎች
COM_ADMIN_ALPHABETICAL_INDEX Alphabetical Index	የፊደል መረጃ ጠቋሚ
COM_ADMIN_CACHE_DIRECTORY (Cache Directory)	(መሸጎጫ ማውጫ)

Figure 3: Translation Component Value Entry

Another part of localization for Joomla is the concept of Virtual keyboard. Various JavaScript virtual keyboards have been created and are available, allowing users to enter their data in local languages. Converting these Javascript virtual keyboards into Joomla module for making use of it on input fields is needed. A virtual keyboard module is developed based on the layout created by Wyvern [9] which is a website providing Java script codes to be customized for adding a graphical keyboard interface to text fields, password fields and text areas so that they can be filled with mouse only.

The script is developed for English language and other foreign languages to enable users enter text using the virtual keyboard. The Java script interface is customized and embedded into a developed module to provide the necessary functionality for Joomla input fields. The customization flow for the emergence of the virtual keyboard is first to import the script and style sheet that provide the functionality and layout of the English virtual keyboard including other foreign languages into html file. The next step is to remove the support for other languages and add a support for Amharic to be included in the language list that will be converted into Joomla module so that it can be imported as a package.

The layout is customized based on a Keyboard standard for Ethiopia prepared by the working group under the direction of technical committee for Information Technology, (QSAE/TC8), and published by the Quality and Standards Authority of Ethiopia (QSAE) [10]. Figure 4 shows the virtual keyboard developed in this work and Figure 5 shows a backend web page of the localized Joomla.

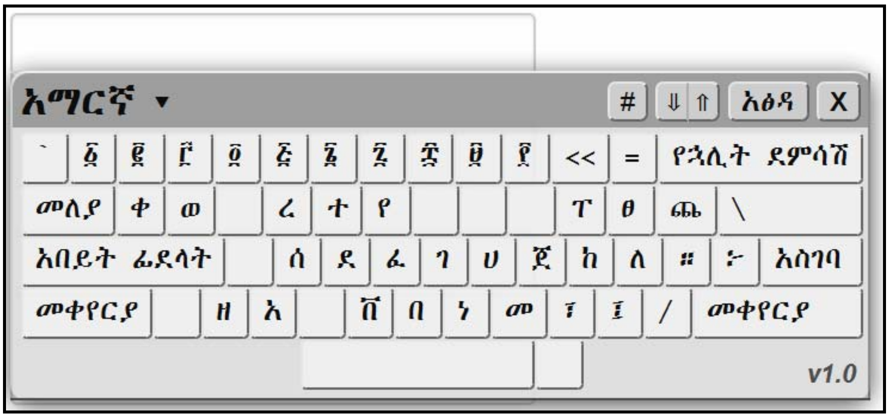

አማርኛ ▾
#
⇓ ⇑
አፅዳ
X
` ፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ << = የኋሊት ደምሳሽ
መለያ ቀ ወ ረ ተ የ ፐ ፀ ጨ \
አበይት ፊደላት ሰ ደ ፈ ገ ሀ ጀ ከ ለ ። ፦ አስገባ
መቀየርያ ዘ አ ሸ በ ነ መ ፣ ፤ / መቀየርያ
v1.0

Figure 4: Amharic Virtual Keyboard Layout

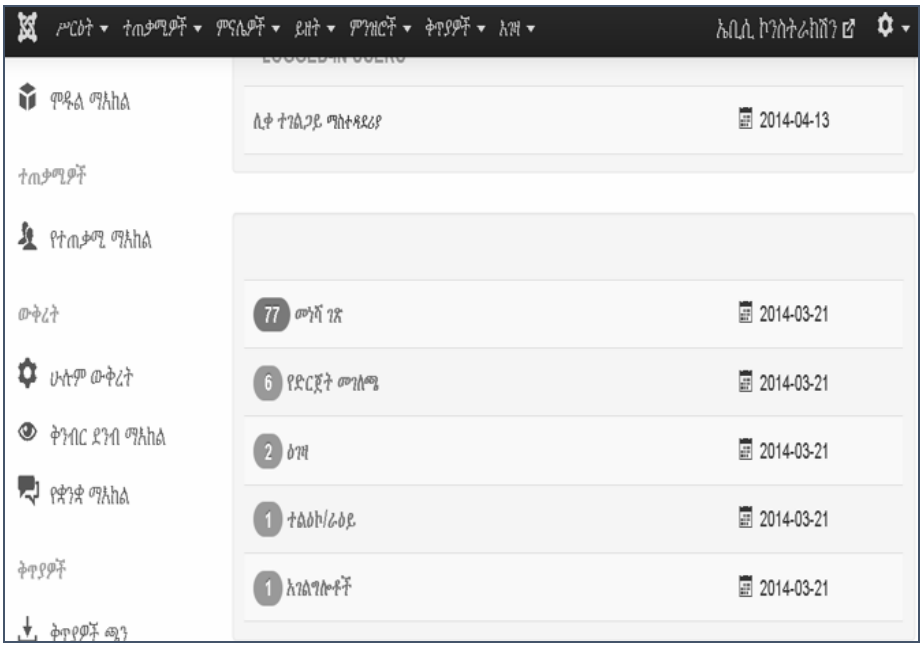


Figure 5: Translated Interface

4. Discussion

The findings of this research basically cover the development of a translation component that enables language translation for several languages, a virtual keyboard design which allows users to enter Amharic characters and an Amharic language pack for Joomla content management system.

The outcome of this research is creating a language pack and virtual keyboard in addition lay down an approach based on the translation component to enable users to create language packs for Amharic, Oromiffa and Tigrigna languages using an easy to use interface which can be easily embedded into the newer version of Joomla.

5. Conclusion and Future Work

In conclusion, the research covers areas of localization for CMS. Different criteria are considered for evaluating and selecting the top performing CMS products. After evaluating the CMS products, selected users are required to fill a questionnaire that assesses experience, popularity of CMS in the local context. The questionnaires are analysed and the CMS with the highest percentage, Joomla, is selected to be localized.

As part of the localization, a language pack for the CMS was created to convert the frontend and backend interface of Joomla. The language pack is designed in a way to be imported into Joomla and users can easily change the language. In addition, virtual keyboard module is designed for Joomla to enable users to encode Amharic text. The keyboard can be imported into Joomla and users can input characters on input fields.

A Translation Component that enables users to translate to Ethiopic languages for Joomla CMS was developed. The translation component enables users to create Joomla language pack for Amharic, Oromiffa and Tigrigna. In order to validate and assess the usability of the developed approach, the localized Joomla is provided to four selected users for the purpose of Website development. The selected users are able to develop the website with relative ease.

In the future, we plan to extend the proposed approach to localize it to all Ethiopic languages based on users' preference. The work focused on specific sets of languages and filling the gaps in between the strings available in a file structure and the interface displayed. The idea can be expanded for full support of other Ethiopic languages.

References

[1] St. Laurent and Andrew M, Understanding Open Source and Free Software Licensing, O'Reilly Media, 2008.

[2] Gillian Klee, "Web Content Management and SharePoint", white paper, Waterstons, England, 2010.

[3] Central Statistical Agency, "Statistical Tables for the 2007 Population and Housing Census of Ethiopia: Country Level", 2007, retrieved from http://www.csa.gov.et/index.php, last accessed on Sept 12, 2013.

[4] Nepa Linux, Guide to Localization of Open Source Software, Centre for Research in Urdu Language Processing (CRULP), National University of Computer and Emerging Sciences (NUCES), Pakistan, 2006.

[5] Zemen Adego, "Open Source Content Management Software Customization for the Ethiopian Government", Unpublished MSc Thesis, Addis Ababa University, 2004.

[6] Lielet Getachew, "Localization of Open Source Content Management Software into Amharic Language", Unpublished MSc Thesis, Addis Ababa University, 2011.

[7] "Making a language pack for Joomla" retrieved from http://docs.joomla.org/J3.1:Making_a_Language_Pack_for_Joomla#How_to_create_localized_CORE_language_packs, last accessed on Jan 13, 2014.

[8] ICT Development Office and Ethiopian Languages Research Center (ELRC), "Translation of ICT Terms into three major Ethiopian Languages Amharic, Oromo and Tigrigna" retrieved from http://www.scribd.com/doc/116396965/ICT-Glossary-Amharic-Tigrigna-Oromifa-Languages, last accessed on Jan 02, 2014.

[9] Brian Huisman, "Java Script Graphical/Virtual Keyboard Interface", retrieved from http://www.greywyvern.com/code/javascript/keyboard, last accessed on Mar 28, 2014.

[10] Technical Committee for Information Technology, Ethiopic Keyboard Layout 1st edition, 2007, Quality and Standards Authority of Ethiopia (QSAE).