OPEN SOURCE CONTENT MANAGEMENT SOFTWARE CUSTOMIZATION FOR THE ETHIOPIAN GOVERNMENT

BY: ZEMENE ADGO

A PROJECT WORK SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER'S OF SCIENCE (M.SC.) IN COMPUTER SCIENCE

ADVISOR: DAWIT BEKELE (Ph.D.)

JUNE 2004
ADDIS ABABA

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. Dawit Bekele, for his continuous guidance, encouragement, and excellent advice throughout the course of this project. Without his help, this work would not be possible.

I also extend my gratitude to Dr. Solomon Atnafu who edited the workplace of OpenCms translations.

Abstract

Most Ethiopian government organizations maintain and exchange documents traditionally. This method of document management is time consuming, costly and unsecured. However, managing documents electronically, greatly improves the internal operations of an organization, raises community satisfaction, and simplifies document management and exchange. In this project, OpenCms which is one of the open source content management software is adapted to enable the government organizations to manage their documents.

Language is another barrier for the growth and development of Information and Communication Technology (ICT) in the country. This is because most software systems are developed in English language but a small percentage of the population understands that language very well. However, by localizing a software product to the country's locales, i.e. languages, cultural norms and standards as well as needs and expectations, the users can interact with the software in the settings that feel natural to them. In this project, OpenCms is localized to Ethiopian locales.

Open source software are appropriate technology for Ethiopia. This is because they can be acquired free of charge and allow users to redistribute, modify and resell the software. However, they are not widely used in the country, there are

many reasons for this some of them are piracy of proprietary software, lack of awareness about open source software and shortage of open source software experts.

Table of Contents

List of Figures

List of Annexes

Acronyms

The following acronyms are used in the document.

BSD	Berkeley Software Distribution
CMS	Content Management System
FS	Free Software
FSF	Free Software Foundation
GNU	GNU's Not Unix (a project to create OSS operating system)
GPL	GNU General Public Licence
LGPL	Lesser General Public License
MIT	Massachusetts Institute of Technology
OSI	Open Source Initiative
OSP	Open Source Project
OSS	Open Source Software
QPL	Q Public License
TCO	Total Cost of Ownership
WTO	World Trade Organization

1. General Introductions

1.1. Introduction

The world is undergoing a revolution in Information and Communication Technologies (ICT) which has major consequence for the socio-economic development of a country.

However, there are countries like Ethiopia which are marginalized from the benefits of ICT for the development of their economy. One of the reasons is that the majority of the populations of these countries use their own unique language and scripts but most ICT hardware equipment and software as well as the content of the Internet are based on English or other Western languages.

Another challenge for developing countries is the price of computer software. In many cases the annual GDP per capital of these countries is lower than the price of single software. Thus users of these countries who managed to buy hardware can't really buy the software they need. As a result, the digital divide between the poor and the rich gets wider.

Nevertheless, there are opportunities that can help theses countries to overcome these problems. Nowadays open source software are widely used around the world. Not only they are acquired without license fee and come with the source code they can be also easily adapted to one's need.

The main objectives of this project are two folds. First, to develop a workable multilingual e-government on-line document management platform development system that helps the Ethiopian government to maintain and manage their Intranet and Internet websites using local languages. Second, to study the applications of open source software for Ethiopia. It also endeavors to promote and create awareness to use open source software in general and content management software in particular in the country.

1.2. Statement of the Problem

The development of a country could not be achieved without good government administration. To improve the efficiency of the administration, the Ethiopian government has laid decentralization system at Woredas level, which empowers them in decision-making process.

Documents are crucial for the activities of all governments including the Ethiopian government organizations. Each government organization has a lot of documents such as annual reports, minutes, letters and budget plans. Most documents are paper based and kept in record office. This is creating major burdens on the record offices of government organizations since they do not have sufficient space to store documents. In addition, documents are kept in very narrow room, which makes searching of particular information a very tiresome and time consuming task. The current traditional document management system has many problems in relation to cost and security. It is not cost effective for collaborative document authoring, editing and reviewing. In addition, the high volume of documents needs big investment on buildings, shelves, folders and papers. Documents security is not guaranteed; documents are vulnerable to fire and other natural disasters as well as other criminal causes. Since there are no backups, once a document is vanished it is less likely to be retrieved.

Nevertheless, increasing number of government documents are produced electronically since computers are increasingly used for word processing purposes. Unfortunately, the electronic version of the document is not stored properly. For this reason it is not easily accessible whenever needed.

One way of solving these problems is using content and document management systems. Document Management is the process of managing documents through their lifecycle, from inception through creation, review, storage and dissemination all the way to their destruction. Content management systems enable organizations to effectively and efficiently administer, handle, maintain, deploy and deliver online content using their Intranet.

In short, digitizing the paper documents and Managing and controlling the ever-increasing electronic copies of documents are crucial for the performance of government organization. It simplifies document searching to get information, raises community satisfaction and saves organizations' time, effort and cost.

Most computer software are developed for use with the most widely used western languages such as English, French and Spanish. However, in Ethiopia, these languages are understood properly by very small proportion of the population. Moreover, the official working languages of the federal as well as regional governments are local languages, which use local scripts different from the Latin alphabets. Therefore, localization of computer software is very crucial in the Ethiopia so that the country benefit from Information and Communication Technology (ICT).

There are three options to develop content management systems as well as other software systems. These are:

- Developing from scratch
- Proprietary software approach and
- Open source software approach

Developing software from scratch is a very time consuming and expensive task. Time and effort are wasted in re-developing an already existing program.

The time which is wasted in re-developing the software can be used for improvement of the software if a developer bases his/her work on the exiting software. This is possible using open source software.

Proprietary software are products which have licenses that restrict the users form copying, distributing, modifying and improving the software. Proprietary software are acquired in binary form but without source code. Using proprietary software has a lot of problems some of them are:

- ✍ Proprietary software may lead to monopolies through lock in and unfair business practices
- ✍ None of the users can adapt or fix errors of the program.
- ✍ Developers cannot learn from the program, or base their new work on the software.

Open source software enable users to access the source code. They do not have license fees and allow users to redistribute, modify, resell, etc the software to somebody else. Open source software have a number of advantages over proprietary software. Some of them are:

- ✍ They have much lower price than proprietary software
- ✍ They can be easily localized to one's language and culture
- ✍ They avoid wasting of effort required for re-developing software that already exists.

Despite, the enormous benefits of using open source software over traditional proprietary software, due to various reasons, they are not widely used in Ethiopia. However, open source software are applicable on the whole to developing countries and specially to Ethiopia.

1.3. Methodology

The requirements of the system were identified based on primary and secondary data. Primary data were collected by interviewing different government officials. Secondary data were assessed by reviewing books, documents found on the Internet and literatures written by different scholars. In addition, several open source content management system software were studied and organizations which use proprietary CMS were also visited. The requirement analysis document was prepared using Unified Modeling Language (UML)

To develop CMS there are three choices: developing from scratch, using proprietary software or using open source software. Because of the requirements of the system and the benefits of using OSS over proprietary software or developing from scratch, for the implementation of the system, OSS were selected.

In Ethiopia, little is heard about open source software. To learn and prepare a paper about open source software different documents were reviewed. Since there are little OSS books and literatures in libraries of Addis Ababa University, most of the documents reviewed were found on the Internet.

A workshop was organized to present open source software and content management system to the community. The presentation was intended to create awareness about OSS to the public

2. Review of Literature

2.1. Open Source Software

Open source software (OSS) movement has caught the attention of millions of users and thousands of companies in the world. The concept of OSS that promote free use and redistribution of software products and their source code attracts more and more users and developers around the world.

In this section of the document the brief history, definition, advantages and drawbacks, and applications of OSS in Ethiopia are summarized.

Review of open source content management systems is done by Frezewd Lemma.[1]

2.1.1. History of OSS

In the early days of computing, computer programs were often shared among developers, just as OSS practitioners do now. During this period Unix operating system was developed by AT&T researchers, and distributed as source code (with modification rights) for a nominal fee. However, as years progressed, software developers, including the Unix system, increasingly closed off their software source code from users.

[1] Frezewd Lemma is a second year post graduate student at Computer Science department of Addis Ababa University

Richard Stallman, a researcher at the MIT Artificial Intelligence Lab, found this closing of software source code intolerable. In 1984 he started the GNU project to develop a complete Unix-like operating system which would be Free Software. In 1985, Stallman established the Free Software Foundation (FSF) to work to preserve, protect and promote Free Software.

In 1991, Linus Torvalds began developing a small operating system kernel called “Linux”, at first primarily for learning about the Intel 80386 chip.

The Linux kernel was combined with the already-developed GNU operating system components and some components from other places and the new systems were called GNU/Linux systems or simply Linux systems,

In 1997, a group of leaders in the Free Software community gathered and coined the term new term open source software as an alternative to free software, which was too confusing and unhelpful (for example, many incorrectly thought that the issue was having no cost).

2.1.2. Free Software

This term was formed by Richard Stallman. Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. Access to the source code is a precondition for this. A program is free software if users have all of these freedoms.

Users do not have to ask for permission or pay to run, copy, distribute, study, change and improve the software.

The above philosophy is legally embedded in a software license called GPL (GNU General Public License) and dubbed Copyleft. The GPL explicitly grants all the above freedoms.

2.1.3. Open Source Software

The Open Source Initiative (OSI) tends to differ with FSF philosophy. OSI explicitly rejects the political and social philosophy of Richard Stallman. Instead it emphasizes on the practicality and technical superiority of a method of software development. When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves.

The distribution terms of open-source software must comply with the following criteria:

- Free redistribution: the license shall not restrict any party from selling or giving away the software
- Source code: the program must include source code, and must allow distribution in source code as well as compiled form.
- Derived works: the license must allow modifications and derived works distributed under the same license of the original software.
- Integrity of the author's source code: the license must explicitly permit distribution of software built from modified source code.

- No discrimination against persons or groups: the license must not discriminate against any person or group of persons.
- No discrimination against fields of endeavor: the license must not restrict anyone from making use of the program in a specific field of endeavor.
- Distribution of license: the rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- License must not be specific to a product: the rights attached to the program must not depend on the program's being part of a particular software distribution.
- License must not restrict other software: the license must not place restrictions on other software that is distributed along with the licensed software.
- License must be technology-neutral: no provision of the license may be predicated on any individual technology or style of interface.

2.1.4. Pros and Cons of Open Source Software

Benefits of OSS

- Avoids lock in: OSS prevents users from becoming dependent on specific program and a specific vendor.
- Low selling Price: open source has a much lower price than proprietary software.

- ✍ Independence: OSS users have more independence from software companies. Even if a software company goes bankrupt, the community still has the source code.
- ✍ Customization: users may modify the source code to meet their specific needs. Beyond customization, anyone can make significant changes to open source code without asking or paying the company which develops the software.
- ✍ Security: OSS is more secure than proprietary software.
- ✍ Transparent: any programmer can see and inspect the source code of OSS.
- ✍ Freedom to control the system: OSS gives users the freedom to control their own computers but with proprietary software the computer does what the software owner wants it to do.

Drawbacks of OSS

OSS can be a long-term viable solution with significant benefits, but there are issues and risks associated with OSS

- ✍ Weak GUI: usually, open source software have a relatively weak graphical user interface (GUI) than proprietary software.
- ✍ Fragmentation: code base may “fork” (be split) into multiple projects, which could lead to incompatible versions.
- ✍ Lack of compatibility: it's difficult to integrate open source & proprietary software.

- Dead-end software: open source projects may fail to gain – or retain large numbers of coding volunteers, which could lead to a dead project.

2.1.5. Motivations of developers

Why do developers contribute to OSS project? Programmers may involve in OSS because the derived personal satisfaction they capture form their work outweighs the costs. The desire for professional recognition and praise may also motivate an individual programmer to solve software products. Other motivations for the participating in OSS project development are:

- Intellectual stimulating : learning and fun
- Code should be open: believers
- Non work functionality: hobbyists, learning and fun
- To sell support: by giving away a product, developers expect to get far more support contracts
- Filling an unfilled demand: this is the primary reason that Open source projects start. It is perhaps not the primary reason why an Open Source project continues, but it explains the origins of Perl, Linux, and Apache.

2.1.6. Intellectual Property

Opponents of OSS argue that open source threatens intellectual property rights. This is a disingenuous argument intended to scare people from learning more about open source. It's inspired by the most radical open source license: the GPL. Source code released under the GPL can't be included in closed, proprietary software. But any content created using open source software (including GPL software) still belongs to the author. For example, authors who create documents in OpenOffice still have copyrights on their work. With open source, only programmers can lose their intellectual property rights. They are willing to give up exclusive ownership of their code to contribute to the community and benefit from community-created programs. Authorship is still respected and rewarded.

2.1.7. OSS Licenses

There are many licenses which qualify the open source/ free software definition and certified by the Open Source Initiative (OSI). Among them the following are the most commonly use licenses.

- ✍ **GNU General Public License (GPL):** The GPL, created by the Free Software Foundation (FSF), is the most important open source license as many open source software are distributed under its terms.

  The GPL permits unlimited free use, modification, and redistribution of software and its source code, but imposes three key restrictions on every licensee:

  - If the licensee redistributes any code licensed under the GPL, it must guarantee availability of the code for the entire work for unlimited replication by anyone requesting it.

- If the licensee redistributes GPL code, it may not charge a licensing fee or royalty, but may charge only for distribution costs.
- If the licensee includes any GPL code in another program, the entire program becomes subject to the terms of the GPL. This is called “viral” clause.

✍ **GNU Lesser General Public License (LGPL):** This license is more or less the same as the GPL except for one important feature: it permits linking with non-free modules. It was originally designed for standard libraries to speed up the adoption of free software since such licensed libraries provide an opportunity for proprietary software to run in a free software system. For this reason it is sometimes also called 'GNU Library General Public License'.

✍ **MIT License:** The MIT license does not really restrict the software or its handling. The only condition is to include the copyright and permission notice in all copies.

✍ **Q Public License (QPL):** QPL prohibits development of proprietary software based on the software licensed under the QPL. Anyone can make modifications and redistribute them in the form of patches along with original source code, as modifications must be distinct from the original. Generated binaries are allowed to have the same name as the original, which is important for dynamic libraries and similar components. Additionally, it forces the author of modification to grant the original producer the right to distribute the changes also under any other license, e.g. a proprietary one.

- **BSD License:** The license was developed by the University of California at Berkeley. It is very non-restrictive, it is similar to the GPL, but does not require derivative works to be subject to the same terms as the initial BSD License. Under the BSD Licenses, distribution of source code is permitted, but not mandated for derivative works. Programs under the BSD Licenses can be combined with proprietary software.

3. Project Works

3.1. Requirement Analysis Document

The first step in software development process is collecting and identifying the needs of the organization to which the software is developed. In this project the need assessment was done through interview and review of related documents.

The existing manual method of document/content management and exchange system used by the target users of the study has a number of problems. Some of the problems are the following:

✍ Shortage of human resources – one of the critical factors in decentralization process is shortage of resources

✍ Shortage of space – Since there is no enough space in most government organizations, records are kept in very narrow room

✍ Difficulty to search information – the way records are stored and the lack of proper reference methods make finding of a particular information or record a very difficult and time consuming task.

✍ Weak communication – Due to poor communication system, organizations and officials found in remote regions/Woredas often receive outdated information.

✍ Lacks of localization – Most computer software are developed to use by most widely used languages like English and French. However, in Ethiopia the official language of the Federal as well as regional governments is neither English nor other western languages. Therefore, without using local languages it may not be possible to use ITC to facilitate the economic development of the country.

- Costly – Exchanging documents using the existing method is very expensive. In addition, to managing the ever-increasing volume of paper based documents, additional personnel as well as investments on rooms, shelves and folders are required.
- Slow method of information delivery – currently a number of organizations use traditional method of information exchange which uses massagers and post offices. This method of communication is very slow.
- Documents are unsecured – manual method of document management is vulnerable to natural disasters and other criminal activities. Since there are no backups, if a document is lost it will be restored.

Generally the existing method of document management and exchange is cost ineffective, time consuming and unsecured.

The New System

The new system is designed to solve the problems of the existing system. The system is implemented by customizing and localizing OpenCms. It satisfies the following functional requirements

- Document management activities: The system enables authorized users to view, remove, copy, move, browse, search, retrieve, send and receive documents electronically using different languages and scripts found in Ethiopia.

- ✍ Version control: The system supports version control, which allows users to add new versions and to return to prior version
- ✍ Upload and download documents: The system enables users to download documents from a server to their computer and to upload documents from their computer to the server.
- ✍ Archiving (including automatic archiving): When files are unused or outdated they should be moved from main storage to the archive storage device. However, users can still search from archived files.
- ✍ Hyper-linking of related documents and updates: The system facilitates the users to open and read related documents.
- ✍ Maintain documents original format: The system maintains files in their original, native format and provide format conversion tools whenever possible.
- ✍ Check out/check in: The system provides check in and check out facility to a user: When a users check out a document, he or she has the option of "locking" it so that other users can view the document, but cannot make any changes to it. This prevents the problem that may arise when several workers attempts to edit the same document at the same time. When finished, the user checks the document back in, making it available to other users once again.

- ✍ Multilingual support: The system provides multilingual document management and exchange to the users.
- ✍ Searching: The system provides search engine support, this includes:
- ✍ Public information: The system provides common and up-to-date information to the general public using websites.
- ✍ Managing workflow: Report and minutes are the major documents prepared and used by government organizations. The new system enables the users to generate, edit and control workflow of minutes and reports of an organization. The system gives a way to create workflow and designate users for a required task
- ✍ Scheduling Facility: The system provides scheduling facility, which invokes custom action periodically or after a specific time. Such custom action can include removing of expired documents, etc
- ✍ Auditing: The system maintains history of documents (auditing). Document history includes who performed an action, its date and time and the nature of the action itself.
- ✍ Browsing Documents: The system enables to browse all available documents, or browse documents by Software Application
- ✍ Maintain website: The system enable the users to create website, create and edit content, manage links among documents, create templates, keep statistics about page hits, prepare documents for multiple targets and prepare documents using multiple formats.

Peer-review was used to verify the completeness of the requirement analysis document. Professors from computer science department were involved during the review.

The detail is found in annex I.

3.2. Minutes Subsystem

One of the requirements of the new system is automated management of meeting minutes. Meeting minutes are important documents which are created, maintained and used by government and non-government organizations of Ethiopia. This is because, most government organizations conduct meetings regularly, at least once in a month when they want to make important decisions, discuss future plans of the organization or talk about other issues. The decisions made and other important points raised during the meeting are recorded in a minute so that they can be referred later on.

Problems of Existing Method of Managing Minutes

Currently minutes and agendas are maintained manually. They are circulated among attendants of the meeting using human transporters. This method of managing and circulating minutes has a lot of problems. Some of them are the following.

- When minutes are circulated among attendants of the meeting for approval somebody, who is not an attendant of the meeting might see a confidential minute.
- Circulating minutes and the agendas from one person to another person using humans is very slow. When all the attendants of the minute are required to approve the minute, approval process can be very long. The participants of the meeting often attend the next meeting without having read the minute of the previous meeting.
- Since there are no reminders, sometimes, minutes of the previous meeting are forgotten, hence are not written before the next meeting. If the minute is not prepared and approved on time, it might be impossible to conduct the next minutes because the next meeting can be based on the previous meeting.

Generally, the existing method of sending agendas and preparing, circulating and maintaining minutes is unsecured, time consuming and inefficient.

The New Minute Subsystem

Regarding to meeting minute management, the new system is designed to solve the problems of the existing method of circulating, preparing, maintaining and retrieving the minutes. The system enables the chairperson of the meeting to automatically send agenda of the meeting to attendants in advance. It also enables the secretary of the meeting to generate, maintain and control workflow of minutes with very little efforts. The system gives a way to create workflow and designate attendants to a minute.

Minutes are incorporated in the system by modifying the source code of OpenCms. For this purpose a number of classes, templates and script files of OpenCms were modified.

3.3. Translations

One of the major works of the project is to localize OpenCms to Amharic language. OpenCms has multi-lingual support, which enables developers to translate the workplace as well as the on-line help system of the software. To fully localize OpenCms workplace 1177 different words, phrases and sentences have to be translated. Among these about 225 are error messages.

In this project the workplace of OpenCms is fully translated to Amharic. Among 1177 different words, phrases and sentences of OpenCms workplace, I initially translated 600 of them. The translations were edited and verified by Dr. Solomon Atnafu[2].

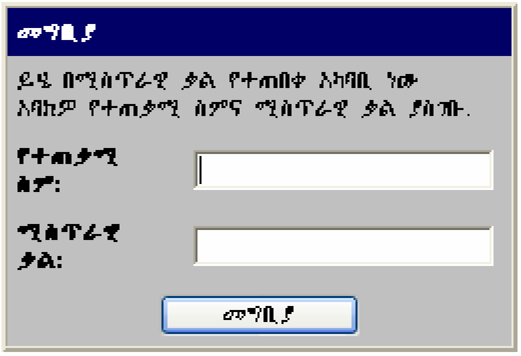
The following steps are important to translate the workplace of OpenCms.

✍ A new module which will contain the localization files should be created

[2] Dr. Solomon Atnafu is a Professor at Computer Science department of Addis Ababa University

- The locale file which will contain the translation words, phrases and sentences should be created. This is a property file. For Amharic language the name the locale file should be workplace_am.properties.

Sample screenshots



ይህ በሚስጥራዊ ቃል የተጠበቀ አካባቢ ነው
እባክዎ የተጠቃሚ ስምና ሚስጥራዊ ቃል ያስገቡ.

Fig.1 Login dialog box

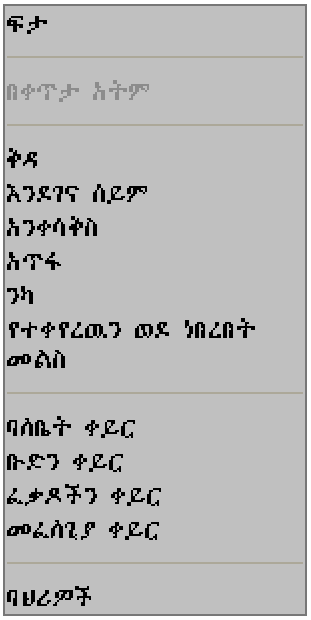
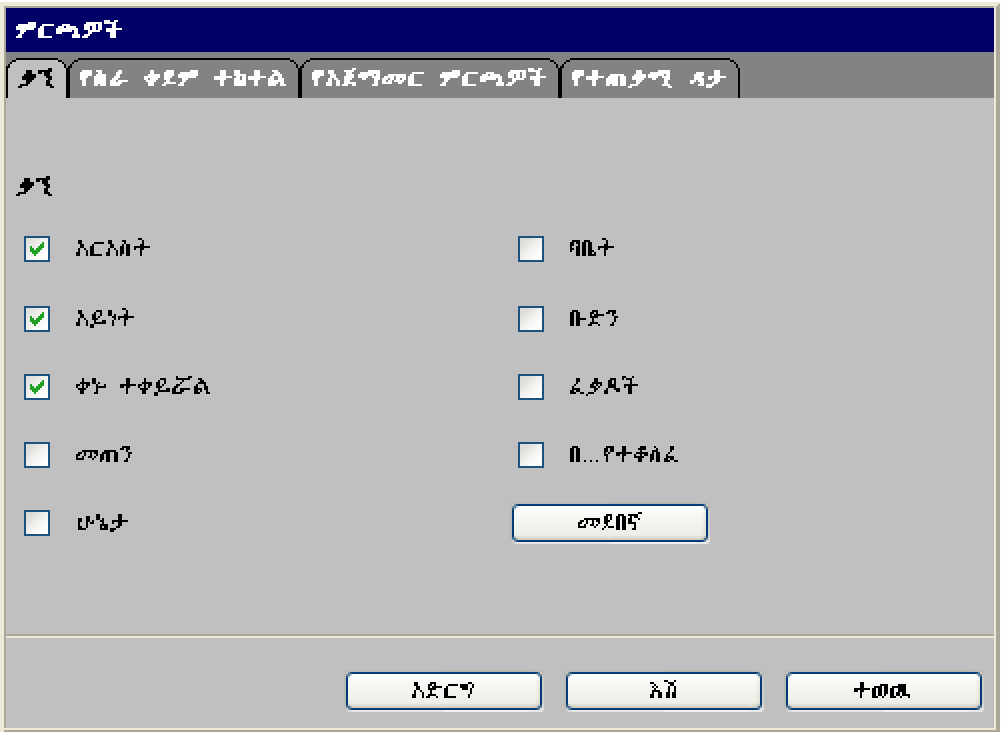


Fig.2 Shortcut menu



ዋና

☑ አርእስት
☑ አይነት
☑ ቀኑ ተቀይሯል
□ መጠን
□ ሁኔታ
□ ባለቤት
□ ቡድን
□ ፈቃዶች
□ በ…የተቆለፈ

Fig.3 Preferences dialog box

The detail is found in annex III.

3.4. OSS in Ethiopia

As it is indicated on the previous chapters, one of the main objectives of this project is to study the applications of open source software for Ethiopia. It also endeavors to promote and create awareness to use open source software in general and content management software in particular in the country.

The concept of open source software that promotes distribution of the software without license fee and redistribution of the products with their source code attracts more and more users and developers around the world.

In spite of the various advantage of OSS over proprietary software, most organizations and users around the world still don't use open source software but prefer to use proprietary software products. There are many reasons for this; one of which is wrong perception about open source software.

Most users of third world countries like Ethiopia, who manage to buy hardware, do not want to buy, and more importantly, can't really buy the software they need, because of the huge license fees. For example, the cost of Windows XP Professional is $269.99 and Microsoft Office Professional is $449.99[3] while the GDP per capital in the country is about $100.00 per year; thus the cost of Windows and Microsoft Office are much higher than the annual income of an average Ethiopian. This encourages piracy of proprietary software, which has become a common practice in the country.

Increasing number of experts believe that open source software are the most viable solution not only for countries such as Ethiopia but also for developed countries. This is because they are very cheap compared to proprietary software and they can be easily customized to local languages and cultures.

[3] http://www.amazon.com

However, very few individuals[4] and organizations use OSS in the country. Even those limited users of OSS are concentrated in international organizations[5]. Some of the possible reasons are:

- Historical reasons: The first personal computers were introduced with Microsoft products and users do not want to migrate to new OSS products.
- Weak copyright enforcement: Users pirate proprietary (mainly Microsoft) products without penalty.
- Lack of compatibility: All OSS are not compatible with the existing propriety software.
- Shortage of support: There isn't enough OSS expertise in the country
- Adaptation of Microsoft products: Users are used to Microsoft products and do not want to learn a new way of working.
- Lack of awareness: The majority of the users as well as developers don't know clearly the merits and demerits of open source software.

However, this situation cannot continue forever. Very soon Ethiopia have to enforce copyright laws, it will be a member of WTO in few years. This will create a big chock, since users will have to pay for the software they are using, which is not affordable by most of them, or migrate to OSS in a very short time.

[4] One of the few individuals, who are active in using open source software and localization of open source software to Ethiopian languages, is Mr. Daniel Yacob Mekonnen. He has translated Gnome Desktop Environment of Linux to Amharic. [http://www.amazon.com]

[5] Dr. Dawit Bekele and Daniel Yacob, ICT standards for Ethiopia, March 2004

Therefore, it is preferable for computer users in Ethiopia to migrate to OSS, following their own calendar, before they obliged to migrate by external forces.

Using OSS software in Ethiopia will have enormous advantages:

- ✍ Legality: Illegal use of proprietary software has a negative effect for the country, in the long term, and should be discouraged. Currently, home users, organizations and companies are addicted to illegal use of proprietary software products. It will be more difficult to relieve the country from the dependency on pirated software where it will be obliged to enforce intellectual copyright laws.
- ✍ Cheaper cost: The cost of open source software is very cheap compared to the cost of proprietary software. Users do not pay for the license to use the software; they only pay for the service: modification and implementation.
- ✍ Income generation: It can develop local expertise and benefit from revenues that OSS generates (training, support, etc.)
- ✍ Lock-in prevention: Individuals and organizations that use OSS are free from the risk of lock-in to specific vendors of software product and services.
- ✍ Help to develop local ITC companies: OSS Lowers barriers to enter to ITC market for local developers and companies seeking to offer ICT solutions.

3.5. Testing of OpenCms

The system has been implemented using OpenCms which is one of the most widely used open source content management software. OpenCms was tested to check whether it satisfies the requirements of the system or not. The test results of OpenCms were compared with the test results of another open source content management software called Plone-Zope. Plone-Zope was tested by Frezewd Lemma. Based on the benchmark taken we were able to choose OpenCms over Plone-Zope. One of the reasons of selecting OpenCms was that it is implemented 100% in Java and it uses servlet technology.

The following features of OpenCms were verified.

- Localization: OpenCms supports localization which enables developers to adapt the software to a specific locale i.e. language, standards and cultural norms. OpenCms fully supports Unicode (UTF-8) as a setup option and thus can natively handle contents in international character sets, e.g. Amharic language.
- Workflow and Task Management: OpenCms has features to manage workflow. Tasks can be created in the Workflow view for different groups, priority levels, due dates, preferred users with notification options
- Versioning: In OpenCms modification of pages are archived and when users need they can restore old versions.
- Forum: OpenCms doesn’t support discussion form
- Scheduling: Scheduling mechanism in OpenCms is similar to “cron” program available in Unix.
- Check in/check out: OpenCms provides check in and check out facility to users. When a document is locked others can view the, but cannot modify

it. When finished, users unlock the document, making the changes available toe other users.

- Auditing: The history function of OpenCms enables to track which changes were made when and by whom.
- Online-help: The online-help system of OpenCms provides inexperienced OpenCms users an easy way to learn the functions of the system.
- Scheduling: OpenCms features an integrated scheduling system. With this system it is possible to invoke a custom action periodically or after a specific time.
- Security: OpenCms allows users to secure all or parts of the website with strong encryption.
- Based on Java / XML: OpenCms is written entirely in Java and uses standard servlet technology. The OpenCms page contents are stored in XML files.
- Browser-enabled work environment: The OpenCms work environment is fully browser based
- WYSIWYG editor: Editing pages is easy with OpenCms; a built-in WYSIWYG editor allows editing pages similar to popular office applications. No HTML knowledge is required to edit a page, yet an integrated source code editor allows experts to exert full control over the HTML source.
- Upload/download: OpenCms has feature that enables users to download documents from a server to their computer and to upload documents from their computer to the server.

- Search engine support: Using the OpenCms documents can be searched using their names or contents.

3.6. Architecture of OpenCms

OpenCms is a client server application that can be used in HTTP-based environments such as the Internet.

Layers of OpenCms

- The presentation layer: The presentation layer consists of a web browser (Example, Internet Explorer or Netscape Navigator) that is used to display and navigate through the HTML user interface.
- The logic layer: The logic layer lies on the web server. The web server can be the Apache web server. All Java classes are installed in the servlet environment on the server.
- The database layer: The database contains tables for resource, user and property data. The OpenCms servlet provides the interface to the presentation and the database layer. It uses the HTTP protocol to establish the communication between the client and the application on the server. The database is accessed through the JDBC interface.

The detail is found in annex IV.

3.7. Installing OpenCms

To install OpenCms (version 5.0) the following software are required.

- The Java JDK 1.4 or later
- Servlet / JSP standards compliant container: e.g. Tomcat 4.1.X
- Relational Database Management System: e.g. MySQL, Oracle

The detail is found in annex V.

3.8. Lessons Learnt

The standard OpenCms (version 5.0) source code has more than 380 java classes, 120 templates and 16 JavaScript files.

One of the problems that were encountered when we modified the source code of OpenCms was to identify which of the classes, templates and scripts files need to be changed for a particular purpose (e.g. to incorporate minutes module in the system). Most of the time we got unpaid support for the problems we faced from the general community of users and developers through mailing lists, web sites, and discussion forums.

Some of lessons learnt by modifying the source code of OpenCms are the following:

- We were able to see practically the importance of reusing the existing code. Our effort was not wasted in re-developing software that already exists. We concentrated on developing features unique to our current task, instead of spending our effort and time on re-thinking and re-writing code that has already been developed by others. This reduces development time.

- ✍ We were able to learn from the source code of OpenCms how very large computer software are constructed
- ✍ We learnt how to extend open source content management software when the required functionalities are not incorporated in the original software

4. Recommendations

4.1. Further Work

In this project the workplace of OpenCms is translated to Amharic language. But the on-line help of OpenCms is not translated. Subsequent developers can translate the on-line help to Amharic.

In Ethiopia there are many languages which are used by different Woredas and regions of the country. In this project OpenCms is customized to Amharic language. It could not possible to customize OpenCms to other languages. Further developers can customize OpenCms to the languages that are used by regional governments.

OpenCms lacks time based versioning. For example, if some one wants to display version one from Sunday to Wednesday and the second version starting from Thursday, users should edit the pages manually. Subsequent developers can add such functionality to OpenCms by modifying the source code.

4.2. Conclusions

Managing and exchanging documents electronically are vital for all government organizations of Ethiopian. It simplifies searching of documents, saves the time wasted to exchange documents, reduces the investment cost needed for shelves, folders and papers and convenient for collaborative editing, reviewing and authoring.

Localizing software products is very important for the growth and development of Information and Communication Technology (ICT) for Ethiopia. This is because the country has its own unique languages and cultures but most software products are developed originally in English language and based on western cultures.

Because of the high price of proprietary software compared to the average individual income of developing countries like Ethiopia, the majority of populations can’t really afford to buy the software they need. Therefore, they will be excluded from the digital world. However, there are open source software which can be acquired free of charge and allow users to adapt the software. Hence, open source software are the best opportunity for Ethiopia.

Glossary

Free software

Free software is software that comes with permission for anyone to use, copy, and distribute, either verbatim or with modifications, either gratis or for a fee. In particular, this means that source code must be available. It does not refer to zero-cost software

Open source software

Open Source is a marketing name for Free Software, coined an attempt to overcome the confusion over the word "free" in the English language. Open Source refers to the fact that the source code of Free Software is open to and for the world to take, to modify and to reuse

Public domain software

Public domain software is software that is not copyrighted. Other people can then use the author’s work freely as they see fit. However, copies or modified versions may not be free at all.

Copylefted software

Copylefted software is free software/OSS whose distribution terms do not let redistributors add any additional restrictions when they redistribute or modify the software. This means that every copy of the software, even if it has been modified, must be free software.

Open standard

Open standard is a standard which is documented, available for all to use, and free of charge

Freeware

Freeware software is available at no price, redistribution of software is allowed but not modification and the source code of freeware is available

Semi-free software

Semi-free software is software that is not free, but comes with permission for individuals to use, copy, distribute, and modify (including distribution of modified versions) for non-profit purposes.

Shareware

Shareware comes with permission for people to redistribute copies, but says that anyone who continues to use a copy is *required* to pay a license fee.

Proprietary software

Proprietary software is software that is not free or semi-free. Its use, redistribution or modification is prohibited, or to do that it requires permission.

Total cost of ownership

Total cost of ownership covers not only the selling price of the software, but any other cost (like failure, training, service, any hardware and software upgrades, and updates)that is caused by the software

References

1. Assefa Admassie (Ph.D.). ICT needs in selected woredas in Ethiopia, pilot study, 2002

2. Dawit Bekele [Ph.D.] and Daniel Yacob Mekonnen. ICT standards for Ethiopia, March 2004

3. David A. Wheeler. Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers!, September 2003

4. Davor Cubranic. Open-Source Software Development

   http://sern.ucalgary.ca/~maurer/ICSE99WS/Submissions/Cubranic/Cubranic.html

5. E. S. Raymond. The cathedral and the bazaar, November 11. 1998

   http://www.cath.org/~esr/writing/cathedral_bazaar/

6. E. S. Raymond. Open source software --- a (new?) development methodology, 1998

   http://www.opensource.org/halloween1.html

7. Eric Lee Green. Economics of Open Source Software.

   http://www:bodfux.org/

8. Martin Bruggink. The Status of Open Source Software in Africa, June 2003

9. Mike Banahan and Graham Taylor. Potential Benefits from the use of OSS, March 2003

10. Nathan Newman. The Origins and Future of Open Source Software. White Paper

http://www.netaction.org/opensrc/future/

11. Nico Coetzee. Free- and Open Source Software in Africa, October 25, 2002

http://www.maailma.kaapeli.fi/africa.html

12. Open Source Initiative. The Open Source definition, 1997.

http://www.opensource.org/osd.html

13. R. F. Young. Sizing the Linux market, second edition. White paper, March 5, 1998.

http://www.redhat.com/redhat/linuxmarket.html

14. R. M. Stallman. GNU project.

http://www.gnu.org/philosophy/free-software-for-freedom.html

15. Rishab Ghosh. Free/Libre and Open Source Software, October 2004

http://www.infonomics.nl/FLOSS/

16. Yohanes Nugroho. Current Status on Open Source Movement in Indonesia.

http://www.opensource.org.id/

Annex I

Requirement Analysis Document for Multilingual e-government content management platform development

1. Introduction

Recognizing the shortcoming of centralized form of public administration the government of Ethiopia has embarked an ambitious plan of decentralization that accords an important role to woredas in the planning and decision-making process within the framework of the Agriculture Development led Industrialization; the government has laid a decentralization and empowerment scheme at woreda level.[6]

Information and Communication Technologies (ICTs) can play a key role in the decentralization process. ICTs will be critical inputs in the economic and social development of woredas by improving public administration. The poorer is the country the greater will be the importance of ICTs as other means of communication such as roads, highways, railroads etc are costly as well as slow to build. [Ibid]

Document and content management is one of the areas that ICTs can play a major role in the government of Ethiopia. The existing method of document management and exchange are so traditional that finding documents is very time consuming and tiresome, documents are easily damaged and lost and government officials found in remote areas of the country often receive outdated information. Document Management is the process of managing documents through their lifecycle. From inception through creation, review, storage and dissemination all the way to their destruction.

[6] Assefa Admassie, ICT needs in selected woredas in Ethiopia, pilot study, 2002

Both government and non-government organizations in the country have a lot of paper documents on the shelf. Till today, documents are exchanged among different government organizations using traditional method, i.e. through post office, using public transport, and personal delivery using messengers. This method of document exchange is inefficient, unreliable, unsecured and above all very slow.

Managing and controlling the ever-increasing amount of paper documents electronically empowers the internal and external communication of the organizations, simplifies document access to get information, makes the information available anywhere in the country, raises community satisfaction and reduces costs. In short, digitizing paper documents, storing and exchanging them electronically will save organization’s time, effort and money.

This document contains requirements of multilingual e-government on-line document management platform. The requirements of government organizations are assessed by interviewing different government officials and by reviewing related documents.

In this document we will first describe the background, objectives, scope and purposes of the project. We will then explain problems of the existing system, functional and none functional requirements of the new system and finally functional model of the system.

2. Background

Information and communication technologies bring tremendous opportunities but also challenges for developing countries such as Ethiopia. They bring the promise of making the currently inefficient government administrations more efficient and bring about good governance that is a prerequisite for the dearly needed economic development. ICTs can help the government in getting information and knowledge up to the most remote parts of the country using wireless technologies. They also make the dream of bringing the best health services, education etc. to remote parts of the country using telemedicine, e-learning, etc.

Unfortunately, these are just promises and not yet reality. Except for a few developing countries such as India, most of the developing world is still getting only marginal benefit out of the information society. This is because there are still a lot of challenges that developing countries have to overcome before they can get full benefit from the information society.

It is not sufficient for a country to import and install the ICT infrastructure and get all the benefits ICT can bring. Some of the technologies are not adequate and should be adapted to the local situation of the countries; others are too costly. Even if adequate infrastructure is put in place, qualified professionals are required to make good use of it.

One major area where these countries need to work to adapt the ICT equipment to the local condition is the area of local languages. Generally, most ICT products that are developed in the north are developed for the use with the most widely used western languages such as English and French. These languages are understood by only a small proportion of the population and it is not possible to have a real penetration of the information society in the population unless it is possible to use local languages with ICTs. The use of local languages is especially crucial in countries such as Ethiopia, where the official working languages of the federal as well as regional governments are local languages, that use, a local script different from the Latin alphabet.

Unfortunately, every government of the world has an administration that works differently from other governments' administrations. Therefore, each country has to develop products that are suited for its own environment. This unfortunately poses great challenges for developing countries that have limited human, material and financial resources.

The activities required for the development of the use of local languages in the information society necessitate research and development activities that universities and research centers in these countries do not have. Therefore, one major task of these countries should be to develop their capacity for research and development in this area.

This work is part of the AAU VarsityNet Multilingual e-government online document management platform development project which is one of the two projects of AAU supported by VarsityNet. VarsityNet is one of the three pillars of the African Learning Network (ALN). VarsityNet tries to establish connectivity at universities and related institutions of higher learning and research, and to stimulate the development of content production and information sharing within this environment. The two main areas of focus are E-government and African.

The main objectives of the project are the following:

- Analyzing the needs of multilingual electronic document management and exchange in government organizations of Ethiopia. The problems as well as limitations of the currently used methods of document management and exchange are also analyzed.
- Designing a workable system. The system will be designed using object oriented approach.
- Developing a prototype that shall solve the main problems faced by the current methods of document management and exchange. The system shall be implemented using open source software.
- Promote the use of open source software/free software in Ethiopia

The system will be implemented using open source operating system and open source distributed and multilingual document management software. Open source software are selected because of their advantages in software project in general and their applicability in developing countries like in Ethiopia in particular. Generally open source software have lower price as well as total cost of ownership, are more reliable, more scalable, more flexible and more secured than their counter proprietary software.

The scope of this project is to develop a workable prototype hat provides a web based multilingual content and document management system that will allow different government organizations to exchange and manage information and documents, send and receive e-mail, conduct discussion forums and chatting, archive documents, upload and download files, search documents with their names and contents, schedule tasks, provide report template, manage minutes, maintain history of documents, manage workflow, browse documents and maintain websites.

However, the system may not support complex content management activities such as language translation, optical character recognition, spelling and grammar checking and the interface of the operating system shall not be localized. The first phase of the project will support Amharic, which is the national language of the country, and English language. Other local languages found in Ethiopia will be incorporated in the next phases of the project.

The prototype will be developed by customizing already existing multilingual open source content and document management software.

3. Purpose of the Project

The major goal of this project is to develop a platform for multilingual and multi-alphabet document management system. It should also allow to easily produce government organizations' Internet and intranet web sites. The platform should have the possibility to easily use databases but also enable easy integration of distributed databases. The platform shall be easily configurable by non-IT personnel.

4. The Current System

The various government offices prepare several reports, which are submitted to zonal and regional offices and other sector offices in the woredas. The main type of reports prepared by the sector offices includes weekly, monthly, quarterly, and annual progress report as well as work plans and in some cases annual budget plans. In the course of preparing these plan documents, the offices need information from the different higher and lower level offices and the general public. Some of the communications are more frequent and are made on daily basis while others are done weekly, quarterly, or annually depending on the nature of the issue to be communicated. [7]

The technology used in the process of information exchange and in the preparation of these documents is rather backward and traditional. The role of modern document management and exchange is very much limited in the country. Most documents are hand or typewritten. However in few but increasingly number of offices documents are created by using word processors, spreadsheets, drawing programs, e-mail programs etc.

Various conventional modes of communication are being used; human transporter circulates the documents if they are meant to other units in the same organization. If they are destined for other offices personal delivery using messengers, face-to-face contacts, public transport, post office and postman who rides motorcycle are used to deliver the documents.

[7] Assefa Admassie, ICT needs in selected woredas in Ethiopia, pilot study, 2002

The central repository of documents in almost all government organizations is the record office. The main purpose of this office is to store both active and dormant documents. These documents are shelved in folders kept in big shelves. Each folder is labeled with the type of the documents it contains and the year. If the folder is not enough for all the documents it will be duplicated and each duplication has a copy number. If shelves become full, they will be duplicated in the same room or in another room if it is available. Since duplication of folder and shelves cannot continue forever, at the end they try to get a space by destructing dead documents and aged dormant files. However, for most organizations destruction of documents even those not used for tens of years is impossible since there is neither law nor regulations that allow them to do so. Therefore, almost all organizations have record offices that are so packed that it is virtually impossible to search and find documents.

4.1. Problems of the Current System

The currently used method of document management and exchange significantly reduce the efficiency of service provision by the government offices. One of the most serious impediments to the smooth operation of the civil service is perhaps lack of accountability and effective management system with clear objectives and well-designed procedures, manuals and policies. Lack of regularity and completeness as well as extreme variation in the size of the annual plans and performance reports suggest the absence of uniform standards and guidelines. [8]

One of the critical factors of decentralization process is shortage of human resources. The available manpower at the different offices are very much unsatisfactory. Moreover, the educational qualification of the employee is far from being satisfactory and very few employees are able to use the computer. The process of data collection, storage, processing, transmission or

[8] Assefa Admassie, ICT needs in selected woredas in Ethiopia, pilot study, 2002

communication leaves much to be desired and lead to serious inefficiency and information loss, thereby reducing the efficiency of service delivery. Most offices do not have adequate space in the archive section to store the information. Most rooms where the records are kept are very narrow, which makes it very difficult to locate a particular file. Documents are simply shelved using box files and paper folders. Files are also exposed to damages by rodents and rain. Moreover, there are no proper reference or code numbers or modern filing systems to locate files. In situations where there is an archive, individuals assigned may not have any training on documentation or file handling. [Ibid]

The public offices do not have the necessary know-how or the technology to process the available information analytically. Comparison with previous years or trend analysis are not included in progress or annual reports. Information is not analyzed to make appropriate decisions. The same problems of implementation are listed in every progress report without identifying the root causes and providing lasting solutions. Moreover, the general public and lower level administrations have no access to these documents, hence the chances for challenging the performance of elected officials or administrative officers are limited. [Ibid]

Because of poor communication systems, particularly in the remote woredas of the country higher-level offices often receive outdated, unreliable and unsecured information. The existing system is generally traditional, unresponsive to public needs and demands, highly bureaucratic and non – participatory. Even if, very few government offices have their own website, the information available on these websites is redundant and little valuable. [Ibid]

Language is one of the major barriers to the formation of perfect knowledge societies. Statistics point out that more than 80% of the content on the net is in English. Yet, very few people in Ethiopia speak that language. People, who are unable to read the content, would be excluded from the knowledge-sharing network. In addition, since Amharic is the official language of the county, the different woredas of the country are expected to communicate using Amharic, especially with the central administration, Federal institutions and organizations outside their respective regions. Therefore, developing contents in local language and translation are very crucial.

The current traditional document management system has many problems in relation to cost, time and security. It is not cost effective in for collaborative document authoring, editing and reviewing. A lot of money is spent to print and duplicate documents. In addition, the very high volume of documents need big investment on buildings, shelves, folders and papers.

The high volume of documents makes searching an immense task. Exchanging the documents is also a big problem. Since traditional means like messengers, and post office are used, it is very costly and slow.

Documents' security is generally not guaranteed since many employees have to spend hours looking for files in the record office. The documents are very much vulnerable for fire and other natural as well as other disasters of criminal causes since there is no efficient way to perform backups.

All in all, the current document management system is time consuming, unsecured and is not cost effective.

5. Proposed system

Overview

The proposed system targets mainly the Ethiopian government administration. The main infrastructure on which we hope this system will run is the WoredaNet network that connects the Ethiopian government institutes up to the woreda (district) level. The infrastructure will serve videoconferencing and information and documents exchange between woredas, regions, sector offices and the federal government. This very much state of the art networking project is under implementation and we believe that such content management system will help in better utilizing the expensive network and in improving the efficiency of the government at all levels.

One of the major features of the content management system should be the support of multiple languages and scripts since different languages are used by different Woredas of the country. The constitution of Ethiopia gives the right to the regions to choose their working language. It is therefore necessary for content management system software to support all languages that are to be used by the various woredas.

5.1. Functional Requirements of the System

The proposed system should provide the following functionalities to the users of the system.

- ✍ Document managing activities: The system should enable authorized users to view, remove, copy, move, browse, search, retrieve, send and receive documents electronically using different languages and scripts found in Ethiopia.
- ✍ Version control: The system should supports version control, which allows users to add new versions and to return to prior version

✍ E-mail service: The system should allow users to send and receive e-mail not only using English language but also using other local languages. The mails should be displayed in tree structure with folders

✍ Upload and download documents: The system should enable users to download documents from a server to their computer and to upload documents from their computer to the server.

✍ Archiving (including automatic archiving): When files are unused or outdated they should be moved from main storage to the archive storage device. However, users can still search from archived files.

✍ Hyper-linking of related documents and updates: The system should facilitate the users to open and read related documents.

✍ Maintain documents original format: The system should maintain files in their original, native format and provide format conversion tools whenever possible.

✍ Check out/check in: The system shall provide check in and check out facility to a user: When a users check out a document, he or she has the option of "locking" it so that other users can view the document, but cannot make any changes to it. This prevents the problem that may arise when several workers attempts to edit the same document at the same time. When finished, the user checks the document back in, making it available to other users once again.

✍ Discussion Forum: The system should provide users to conduct discussion forum: One person can post a topic or a question and then others can respond to it.

✍ Multilingual support: The system should provide multilingual document management and exchange to the users.

✍ Searching: The system should provide search engine support, this includes:

▪ Full text search- this lets a user to find documents by their content, i.e., words or phrases in the document

▪ Generating multiple search indices for different languages

  - Search document types like PDF, MS Word or Excel, HTML, XML, Power Point, Word Perfect, JPEG, audio, video and other electronic documents. This means having instant access to information, regardless of the format of the information
  - Finding documents by index fields such as date, file name, file type, and size

- Report template: The system should provide report forms that each generated report should follow. This is used to avoid the variation in size and format of reports that are generated by different government organizations.
- Public information: The system should provide common and up-to-date information to the general public using websites.
- Managing workflow: Report and minutes are the major documents prepared and used by government organizations. The new system should enable the users to generate, edit and control workflow of minutes and reports of an organization. The system should give a way to create workflow and designate users for a required task
- Scheduling Facility: The system should provide scheduling facility, which invokes custom action periodically or after a specific time. Such custom action can include removing of expired documents, etc
- Auditing: The system should maintain history of documents (auditing). Document history includes who performed an action, its date and time and the nature of the action itself.
- Browsing Documents: The system should be able to browse all available documents, or browse documents by Software Application
- Maintain website: The system should enable the users to: - create website, create and edit content, prepare a poll and survey area to collect information, manage links among documents, create templates, keep statistics about page hits, prepare documents for multiple targets and prepare documents using multiple formats.

5.2. Nonfunctional requirements

This section describes other features, characteristics, and constraints that should be satisfied in addition to the above system functional requirements of the system.

✍ The system should be reliable i.e., the system should consistently performs according to its specification.

✍ The system should be available for the office-working hours in given day.

✍ The system should provide features like manual backup as well as automatic backup on regular basis

✍ The system should provide secured method of document exchange and management: the system will assign rights and permissions to users and groups of users based on the roles of the user in the organization. The security should be implemented using user name and password which should be entered by the user when he or she logs into the system.

✍ Each user group should have different interface, which can be customized to ones' need (Personalization). The system should have web based Graphical User Interface (GUI), which is common for most desktop applications, and familiar to most computer users. This style of user interface might minimize the time needed for users to adopt and use the new system. The system should be menu driven with menus, toolbars, buttons and other user interface, which are common to current desktop applications.

✍ The interface should incorporate a button or menu that allows the user to change the language he/she is currently using.

Documentation

The activities and outputs of each system development stage in the project workflow will be properly documented. The documents produced at the end of each stage should be organized and compiled together at the end of the project for future reference, system maintenance & enhancement, and to give support. Other different documentations shall also be compiled and provided at the completion of the project which includes:

- ✍ User training manual
- ✍ A computer based tutorial on how to use the system
- ✍ Complete reference document
- ✍ Compiled help system

Hardware Requirements

The new system will be implemented using the WoredaNet infrastructure, which is currently under construction. This infrastructure is based on the latest versions of computer hardware, and network devices.

Pseudo Requirements

The prototype will be developed by extending and customizing open source document management and content management software and object development technology to simplify implementation, maintenance and future enhancement. The servers should run Linux operating system, which is one of the most widely used open source operating system.

6. System Model

Use case

A use case describes the functionality of the system from the use's point of view. After analyzing the functional requirements of the new system, the following uses and actors are identified.

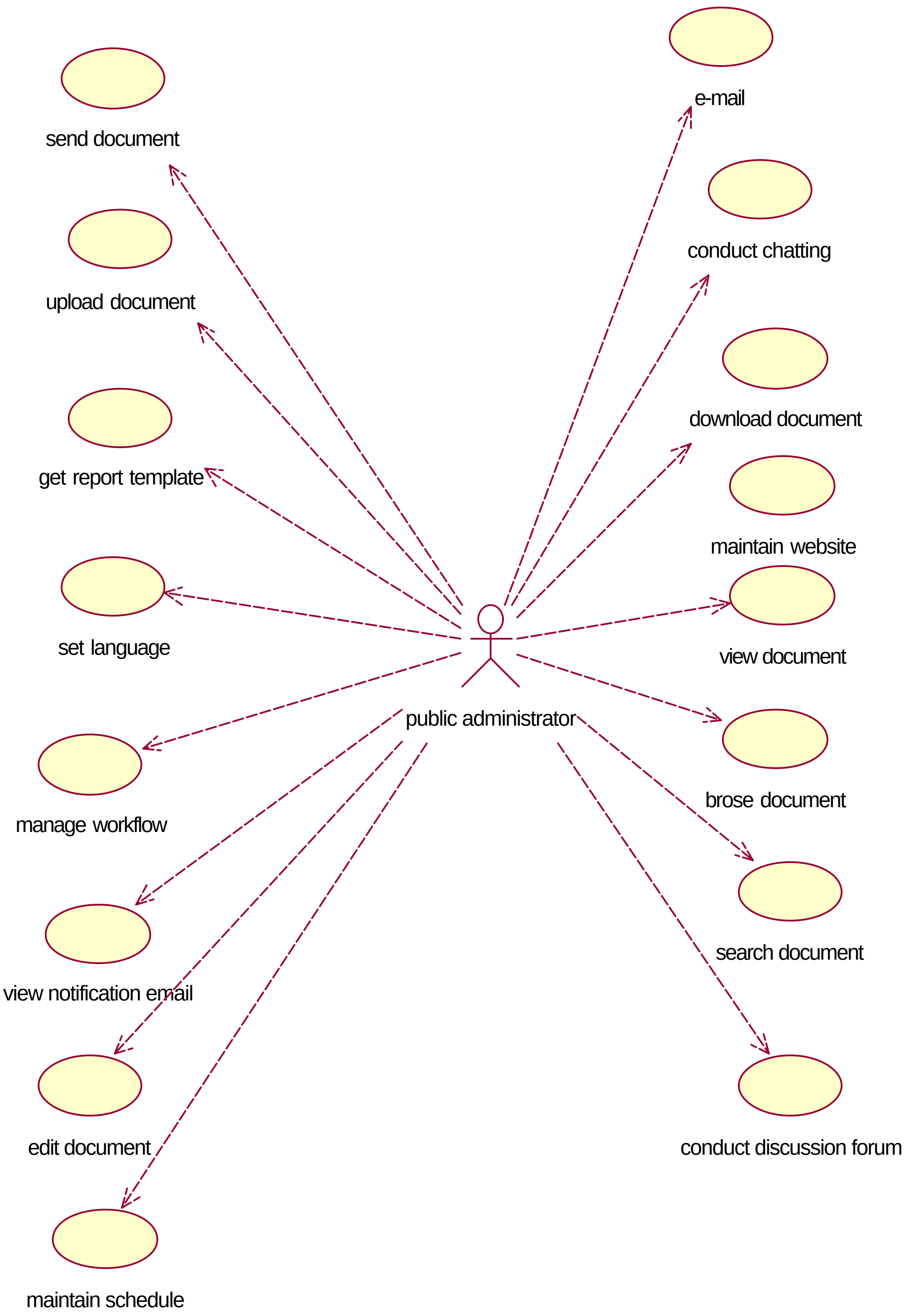
Actors	Use case	Description
Public servant, public administrator, Server Administrator	Upload document	This use case describes the activities done by the system when user wants to puts/loads documents to the repository computer (server) from which other users can review, edit, view, etc the document.
Public servant, public administrator, Server Administrator	Download document	This use case describes the activities done by the system when user wants to brings/takes away documents from the repository computer (server) to his/her computer.
Public servant, public administrator, server administrator	Browse document	This use case illustrates the tasks done by the system when the user wants to display the available list of documents or files that are allowed to the user to see, to edit, to delete etc.
Public servant, public administrator, Server administrator	Search document	This use case describes the activities done by the system when the user wants to locate a document or files, but does not remember the actual location or name of the file. The search can be done by file name, by content, by size and by creation or modified date.
Public servant, public administrator, server administrator	Send document	This use case describes the activities done by the system when the user wants to dispatch/forward documents to the a particular/group of users
Public servant, public administrator, server administrator	Conduct discussion forum	This use case describes the activities done by the system when users want to discuss an issue; they can post their question and receive responses from other individuals.
Public servant, public administrator, server administrator	View document	This use case illustrates the activities done by the system when a user wants to see contents of a document. It can be opened on its native format or converted to other formats if it is supported by the system.
Public servant, public administrator, server	Set language	This use case describes the activities done by the system when a user wants

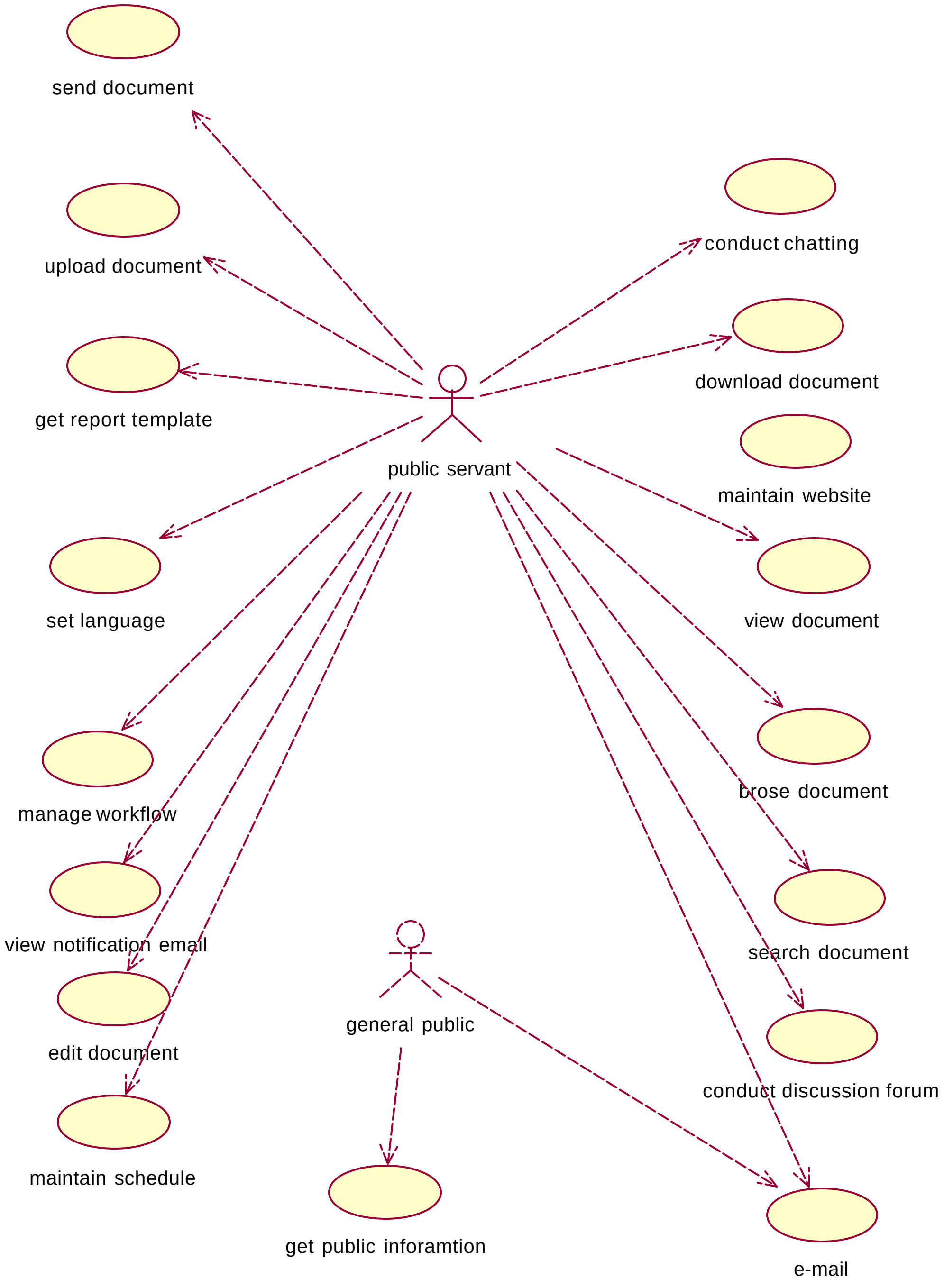
administrator		to change the language from English to Amharic and vice versa. After changing the language the menu bar, toolbars, buttons, dialog boxes, help system etc should be displayed by the selected language. When the user creates and edits a document, the content should be displayed in the selected language.
Public servant, public administrator	Edit document	This use case describes the activities done by the system when a user wants to open and modify the contents of a document or file on its native or converted format using WYSIWYG (What you see is what you get) editor tools.
Public servant, public administrator, server administrator	View document history	This use case describes the activities done by the system when a user wants to know when and who edited, formatted, viewed, deleted, archived and recovered a particular file or document.
Public servant, public administrator, server administrator	Conduct chatting	This use case describes the activities done by the system when users want to carry out online textual conversation with other users.
Public servant, public administrator	View notification e-mail	When the system sends notification email to inform the user about something according to a workflow requirement, the target computer displays notification signal. This use case describes the activities done by the system when a user wants to display the newly arrived email messages.
Public servant, public administrator, server administrator	Maintain schedule	This use case describes the activities done by the system when a user wants to schedule tasks like backup, archiving, removal of unwanted and temporarily documents and files on specified time.
server administrator	Archive document	This use case describes the activities done by the system when a user needs to move inactive/dormant document, which are not required currently, from the main repository to the archive storage.
Server administrator Public servant, public administrator	Maintain website	This use case illustrates the following activities - Authoring content - Managing Links - Creating templates - Uploading Graphics and Multimedia - Publishing content on the website - Preparing documents for multiple targets - Creating mailing lists - Creating polls and survey area - Keeping statistics about page Hits

		- Providing information to the general public
General public	Get public information	These use case describes the activities done by the system when a user wants to display common information to the general public.
Public servant, public administrator	Get report template	This use case describes the activities when the user wants to get standard report format for a particular type of report
Public servant, public administrator, general public, server administrator	E-mail	This use case describes the activities done by the system when the user wants to send, receive, and open e-mail message
Public servant, public administrator	Manage workflow	This use case describes the activities done by the system when the user creates workflow (multi step job needs to be performed by many people), assigns a task to a person. The system prepares report according to the work done by the assigned users. For example for managing minute circulation for approval, sending and receiving reports among users.

Table 1. Description of use cases

Use Case Diagram




send document
upload document
get report template
set language
manage workflow
view notification email
edit document
maintain schedule
public servant
conduct chatting
download document
maintain website
view document
brose document
search document
conduct discussion forum
e-mail
general public
get public inforamtion

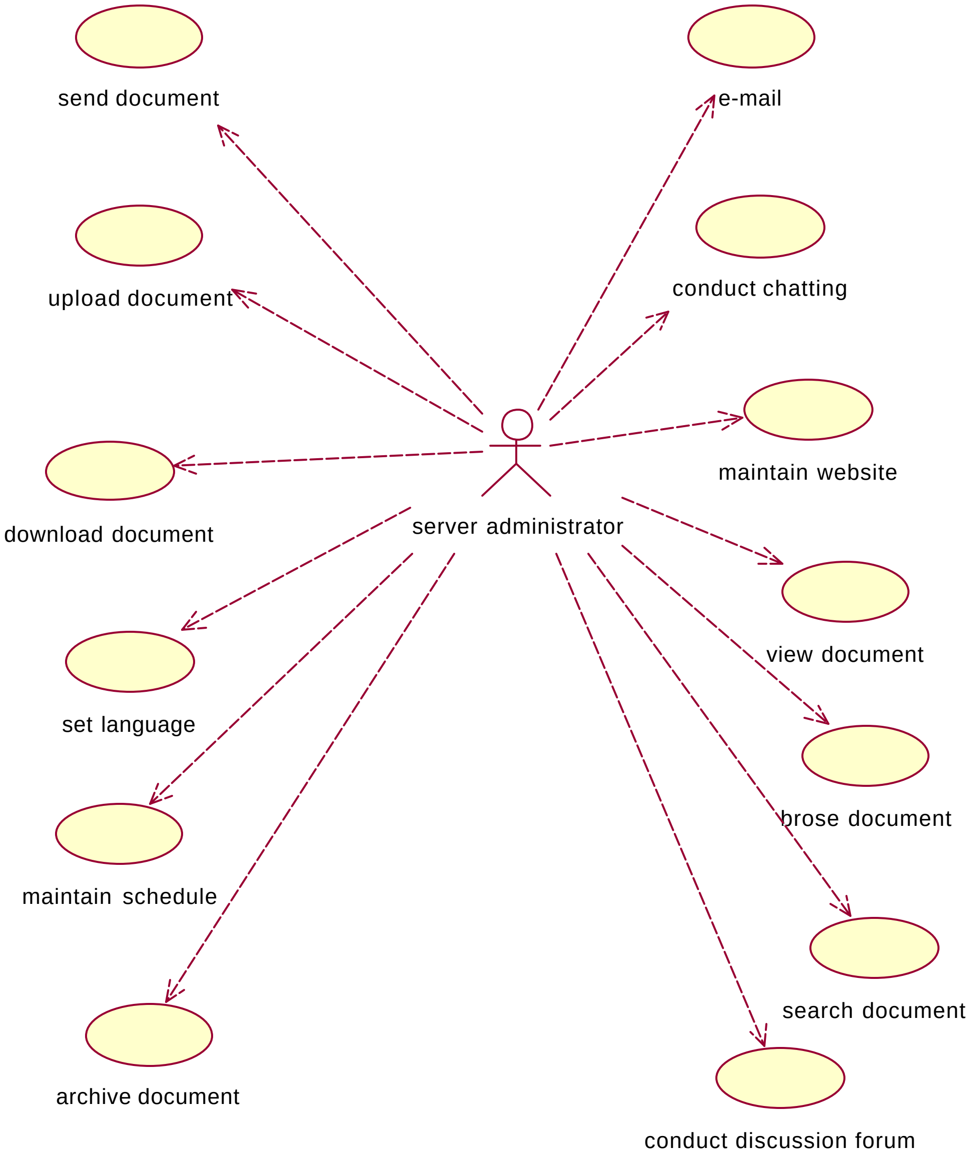


Fig. 1. Use case diagram

Annex II

Open Source Software

1. Introduction

Open source software (OSS) movement has caught the attention of millions of users and thousands of companies in the world. In fact it is becoming the "buzz word" of today. The concept of OSS that promote free use and redistribution of software products and their source code attracts more and more users and developers around the world.

However, many organizations and users still don't use open source software. They rather prefer to use proprietary and expensive software products. There are many reasons for this; one of which is wrong perception about open source software that this document tries to eliminate.

The objective of this document is to describe what open source software are, how they are developed, what are their benefits and drawbacks of open source software, etc. It also describes the relevant of open source software in developing countries such as Ethiopia.

2. History

In the early days of computing (approximately 1945 to 1975), computer programs were often shared among developers, just as OSS practitioners do now. During this time period Unix operating system was developed by AT&T researchers, and distributed as source code (with modification rights) for a nominal fee. However, as years progressed, and especially in the 1970s and 1980s, software developers increasingly closed off their software source code from users. This included the Unix system itself; many had grown accustomed to the freedom of having the Unix source code, but AT&T suddenly increased fees and limited distribution, making it impossible for many users to change the software they used and share those modifications with others.

Richard Stallman, a researcher at the MIT Artificial Intelligence Lab, found this closing of software source code intolerable. In 1984 he started the GNU project to develop a complete Unix-like operating system which would be Free Software (free as in freedom, not as in price). In 1985, Stallman established the Free Software Foundation (FSF) to work to preserve, protect and promote Free Software; the FSF then became the primary organizational sponsor of the GNU Project. The GNU project developed many important software programs, including the GNU C compiler (gcc) and the text editor emacs. A major legal innovation by Stallman was the GNU General Public License (GPL), a widely popular OSS software license. However, the GNU project was stymied in its efforts to develop the “kernel” of the operating system. The GNU project was following the advice of academics to use a microkernel architecture, and was finding it difficult to develop a strong kernel using this architecture. Without a kernel, the GNU project could not fulfill their goal.

Meanwhile, the University of California at Berkeley had had a long relationship with AT&T’s Unix operating system, and Berkeley had ended up rewriting many Unix components. Keith Bostic solicited many people to rewrite the remaining key utilities from scratch, and eventually managed to create a nearly complete system whose source code could be freely released to the public without restriction. The omissions were quickly filled, and soon a number of operating systems were developed based on this effort. Unfortunately, these operating systems were held under a cloud of concern from lawsuits and counter-lawsuits for a number of years. Another issue was that since the BSD licenses permitted companies to take the code and make it proprietary, companies such as Sun and BSDI did so - continuously siphoning developers from the openly sharable code, and often not contributing back to the publicly available code. Finally, the projects that developed these operating systems tended to be small groups of people who gained a reputation for rarely accepting the contributions by others (this reputation is unfair, but nevertheless the perception did become widespread). The descendents of this effort include the capable operating systems NetBSD, OpenBSD, and FreeBSD, as a group called the BSDs. However, while they are both used and respected, and proprietary variants of these (such as Apple Mac OS X) are thriving, another OSS effort quickly gained the public interest and much more market share.

In 1991, Linus Torvalds began developing a small operating system kernel called “Linux”, at first primarily for learning about the Intel 80386 chip. Unlike the BSD efforts, Torvalds eventually settled on the GPL license, which forced competing companies working on the kernel code to work together. Torvalds made a number of design decisions that in retrospect were remarkably wise: using a traditional monolithic kernel design (instead of the “microkernel approach” that slowed the GNU project), using the Intel 386 line as the primary focus, working to support user requests (such as “dual booting”), and supporting hardware that was technically poor but widely used. And finally, Torvalds stumbled into a development process rather different from traditional approaches by exploiting the Internet. Torvalds’ new process looked rather different than more traditional approaches. He publicly released new versions extremely often, and he quickly delegated areas to a large group of developers. Instead of depending on rigid standards, rapid feedback on small increments and Darwinian competition were used to increase quality.

When the Linux kernel was combined with the already-developed GNU operating system components and some components from other places (such as from the BSD systems), the resulting operating system was surprisingly stable and capable. Such systems were called GNU/Linux systems or simply Linux systems, In 1997, a group of leaders in the Free Software community gathered. They were concerned that the term Free Software was too confusing and unhelpful (for example, many incorrectly thought that the issue was having no cost). The group coined the term open source as an alternative term, and Bruce Perens developed the initial version of the open source definition to define the term. The term open source is now very widely used, but not universally so; Richard Stallman (head of the FSF) never accepted it.

Major Unix server applications (such as the OSS Apache web server) were easily moved to GNU/Linux system. As a result, GNU/Linux and the BSDs rapidly gained significant market share in the server market. In 2002, the first user-ready versions of capable and critical desktop applications (Mozilla for web browsing and OpenOffice for an office suite) were announced.

3. Free Software

This term was formed by Richard Stallman. Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- ✍ The freedom to run the program, for any purpose.
- ✍ The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- ✍ The freedom to redistribute copies so you can help your neighbor.
- ✍ The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

A program is free software if users have all of these freedoms. Thus, you should be free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission.

It is important to understand that the term 'free' is about freedom and not about price. The definition does not exclude charging money for the distribution of your software Selling Free Software, but the paying customer must still have the described freedom, otherwise it would not be right to call it free software.

The above philosophy is legally embedded in a software license called GPL (GNU General Public License) and dubbed Copyleft. The GPL explicitly grants all the above freedoms, and tops those off with a requirement which makes sure that the use of the code of GPL software does not result in anybody redistributing the same and expanded code with a proprietary scheme.

Free Software is not the same as Freeware, which is Proprietary Software made available at no price, providing various degrees of freedom of use, but in most cases not other freedoms as described by FSF. Microsoft Internet Explorer, for example, is a web browser available at no price.

Richard Stallman said, “The identifying characteristic of Free Software is not the absence of price. Some free software is sold (Red Hat). Some proprietary software is given away (freeware).”
Fee software gives users the freedom to control their own computers with proprietary software, the computer does what the software owner wants it to do, not what you want it to do. And free software gives users the freedom to cooperate with each other, to lead an upright life.

4. Open Source Software

The Open Source Initiative (OSI) tends to differ with FSF philosophy, and instead emphasizes the practicality and technical superiority of a method of software development. It explicitly rejects the political and social philosophy of Richard Stallman. Instead it points to the open source model as a superior model of developing software, which is based on the availability of the source code, combined with rights to inspect, modify and distribute to everyone, resulting in a technically superior software, because any programmer can see and fix the problems and bugs, and improve functionality at a far faster pace than a proprietary model of developing software can ever do. The OSI puts their case as follows: “The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.” The important features of this model are that a large number of programmers can in the age of Internet co-operate on producing software of a very complex nature, despite being physically far apart, and despite not working in a traditional ‘cathedral-building style’ setting.

5. The Open Source Definition

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

The fundamental difference between the free software and open source software movements is in their values, their ways of looking at the world. For the Open Source movement, the issue of whether software should be open source is a practical question, not an ethical one. Eric Raymond put it, ``Open source is a development methodology; free software is a social movement.'' For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution.

Example of open source software

Operating Systems: GNU/Linux, BSDs
General Utilities: GNU utilities
Programming Languages: Python, Perl, Tcl, GNU C/C++
Windows Systems: XFree86
Desktop Environments: GNOME, KDE
Web Browsers: Mozilla
Office Suits: OpenOffice, StarOffice, KOffice
Server Type software: Samba, Apache, Zope, MySQL

6. Pros and Cons of Open Source Software

Proprietary software may lead to monopolies through lock in (Lock in occurs when a user becomes dependent on a specific program from a specific vendor) and unfair business practices.
In addition, proprietary software has the following problems:

- ✍ Fewer people use the program.
- ✍ None of the users can adapt or fix the program.
- ✍ Other developers cannot learn from the program, or base new work on it.

6.1. Benefits of OSS

Organizations can achieve many benefits over traditional proprietary software by using OSS

- ✍ **Selling Price:** Open source has a much lower price. The price of an open source program is usually far less than a comparable proprietary program. Open source software doesn't have to be "no fee" but most programs are.

Users can either download the software directly or pay a negligible fee to have a CD-ROM burned and shipped.

- ✍ **Total Cost of Ownership (TCO):** Total cost of ownership (TCO) covers not only the selling price of the software, but any other cost (like failure, training, service, any hardware and software upgrades, updates that is caused by the software). TCO of open source software are potentially lower. For instance, the software typically has lower hardware requirements than proprietary alternatives. Software can be downloaded from the Internet quickly and conveniently.

✍ **Reliability:** One of the reasons why OSS is more reliable than proprietary software is that free software gets the whole community involved in working together to fix problems. Users not only report bugs, they even fix bugs and send in fixes. Users work together, communicate each other, to get to the bottom of a problem and make the software work trouble-free.

Another reason is that developers really care about reliability. OSS packages do not always compete commercially, but they still compete for a good reputation, and a program which is unsatisfactory will not achieve the popularity that developers hope for. What's more, an author who makes the source code available for all to see puts his reputation on the line, and had better make the software clean and clear, on pain of the community's disapproval.

- ✍ **Independence:** Open source means greater independence from companies. Open source users have more independence from software companies. Even if a software company goes bankrupt, the community still has the source code. This independence also means "end of life" decisions or undesirable new features can't be forced on the users. With open source software, no one can force the end of a product. Users can choose to move to another product if that appears to be the best solution - but they can also choose to maintain the software in question or to join with others to do so. The same is true if product development moves in a direction incompatible with a user's needs: the recipient can choose to move to a different product, but can also choose to maintain the software

- **Efficiency**: open source products have access to extensive technical expertise, and this enables the software to achieve a high level of efficiency. The rapid release rate of OSS distributes fixes and patches quickly, potentially an order of magnitude faster than those of commercial software.
- **Flexibility**: All users may modify the source code to meet their specific needs as long as they continue to comply with the Open Source Definition. Beyond customization, anyone can make significant changes to open source code. Anyone can adapt the software without asking or paying the programmer or a company. Such adaptations lead to innovations which eventually benefit the whole software-using community. Proprietary software may offer new features or flexibility in each new version.

- **Support**: Because OSS vendors compete against one another to provide support, the quality of support increases while the end-user cost of receiving the support decreases. Open source can create support that lasts as long as there is demand, even if one support vendor goes out of business.

- **Security:** Open source is more secure. The transparency and reliability of open source may mean better security. Multiple programmers independently finding and fixing the same program to discover the best solution. Security threats like viruses and worms exploit bugs to damage computers. Eric Raymond explains: One advantage of parallel debugging is that bugs and their fixes are found / propagated much faster than in traditional processes.

  Transparency is sometimes misrepresented as a critical security threat. Transparency may make it easier for a criminal gain access to a program. However, the open source model leverages a community of programmers to maintain and improve security. This collective benefit seems to outweigh the danger of transparency. Also, the security of a system depends much more on careful deployment and maintenance. Proprietary software may include spyware to monitor users and collect data. With open source it's nearly impossible to hide spyware or similar threats to privacy and free speech. No single company can control open source so a monopoly can't form. Open source tries to prevent monopolies, spyware, and other threats, but people must assert their rights as consumers and citizens, not just with software.
- **Transparent**: Software is better when it's transparent. If software is transparent, any programmer can see what happens and why. Proprietary software is not transparent. Most computer users are familiar with the strange error messages that appear when a program fails. Often, these messages use special codes to express the problem. These codes point to secrets without actually revealing what went wrong. Only someone who knows the secrets can understand the codes.

- **Interoperability**: The open source model builds open standards and achieves a high degree of interoperability. While proprietary software typically depends on monopoly support with one company providing support and “holding all the cards” (i.e., access to the code) for a piece of software, the publicly available source code for OSS enables many vendors to learn the platform and provide support.

- **Localization**: Since OSS comes with the source code; one can easily customize existing software to suit one’s language and culture.

- **Re-invention of the wheel**: Since one can reuse existing code, effort is not wasted re-developing software that already exists. This enables Programmers to concentrate on developing features unique to their current task, instead of spending their effort on re-thinking and re-writing code that has already been developed by others. Code re-use reduces development time, provides predictable results and helps the progress of software technology. That’s why, now there is a large amount of OSS already available, powerful and complex applications are being written very quickly.

6.2. Drawbacks of OSS

OSS can be a long-term viable solution with significant benefits, but there are issues and risks associated with OSS

- **Poor code**: Poor code often results if the open source project is too small or fails to attract the interest of enough skilled developers.
- **Weak GUI**: Highly technical, skilled developers tend to focus on the technical user at the expense of the non-technical user. As a result, OSS tends to have a relatively weak graphical user interface (GUI) and fewer compatible applications, making it more difficult to use, in particular, for desktop applications (although some OSS products are greatly improving in this area).
- **Fragmentation**. Code base may “fork” (be split) into multiple projects, which could lead to incompatible versions. As new versions of the OSS are released, organizations need to make sure that the versions to be integrated are compatible, ensure that all developers are working with the proper

version, and keep track of changes made to the software. Without a formal corporate structure, OSS faces a risk of forking of a code base, resulting in the development of multiple incompatible and inconsistent versions of programs. This can occur when developers try to create alternative means for their code to play a more significant role than achieved in the base product.

- **Lack of Integration:** It's difficult to integrate open source & proprietary solutions. Integrating any two programs is often challenging. It may be difficult to integrate some open source and proprietary solutions. This is usually because proprietary companies prefer their customers to use their brand for most or all of a solution.
- **Dead-end software** – Open Source projects may fail to gain – or retain large numbers of coding volunteers, which could lead to a dead project. How does this differ from a company going out of business or deciding to drop support? At least in Open Source the source code is available.

7. Making Money with Open Source Software

Many people claim that open source software does not fit in our economical system because of its distribution terms. Although making profits might be more difficult today with open source software than with proprietary software, recent times have shown that you can earn your living with open source software. A good example is the Free Software Foundation that has existed for quite a while. Besides, there is no proof that the situation will remain in the future the way it is now as a lot of money has been invested in open source software and a certain hype can be observed these days, too.

- **Software Distributions:** Distributors simply sell copies of open source software. This business is based on the idea that the regular user of open source software is willing to pay a small amount for comfortable access to the software.

- **Service:** There are many different services in the Open Source Software. Examples are support, training or simply paid bug fixing.
- **Hardware Manufacturers:** As hardware devices cannot be used without the appropriate software, vendors usually spend a remarkable amount of financial resources on the production of driver software. It is the usual procedure to make this software available for free, but without source code. However, more and more companies also start to participate in open source projects to assure the compatibility and support for their products.
- **Information**: Books, magazines and news services provide required information about open source software for a reasonable price, e.g. nicely printed manuals.
- **Advertisement:** open source products can function as an advertisement to sell services and other goods. For example, Netscape uses Mozilla browser as a free advertising for their other goods and services (primarily their web servers and web portals).

8. Motivations

Why do developers contribute to OSS project? Programmers may involve in OSS because the derived personal satisfaction they capture form their work outweighs the costs. The desire for professional recognition and praise may also motivate an individual programmer to solve software products. Other motivations for the participating in OSS project development are:

1. Intellectual stimulating –learning and fun
2. Code should be open –believers
3. Non work functionality-hobbyists, learning and fun
4. To sell support – by giving away a product, developers expect to get far more support contracts
5. Filling an unfilled demand: This is the primary reason that Open source projects start. It is perhaps not the primary reason why an Open Source

project continues, but it explains the origins of Perl, Linux, and Apache quite well. Many Open Source products were created to fill an unfilled market. For example, Linus Torvalds wanted an inexpensive Unix-like operating system for his 80386-based computer. And better yet if there was source code available for it. There wasn't one. So he started creating one. If there had already been an inexpensive Unix-like product with source available for a reasonable price, it is unlikely that Linus Torvalds would have ever started the Linux project. There are people who like source code. They like to read and modify the source code to suite their needs.

6. Free advertising: If there is no obvious commercial market for a product there will be little or no gain from keeping it proprietary but much gain can be found if it is released as Open Source (in the form of prestige, free advertising, free programming, and others with similar needs can help to enhance and debug the product). For example, Netscape can no longer make money-selling browsers, due to Microsoft's bundling of Internet Explorer with the Windows OS. However, it uses Mozilla browser as a free advertising for their other goods and services (primarily their web servers and web portals).

9. The Cathedral and the Bazaar

The open source development process differs from the traditional commercial proprietary model. Eric Raymond likens the corporate model, whereby a corporation produces and sells proprietary software, to a cathedral and the open source model to a bazaar.

The process of open source software development is like a bazaar of differing agendas and approaches. The open source model relies on a network of "volunteer" programmers, with differing styles and agendas, who develop and debug the code in parallel. From the submitted modifications, the delegated leader chooses whether or not to accept one of the modifications. If the leader thinks the modification will benefit many users, he will choose the best code from all of the submittals and incorporate it into the OSS updates. The software is released early and often.

In contrast to the traditional style of software development, which requires more centralized approach, software carefully crafted [by individuals or small teams] working in splendid isolation, with no beta to be released before its time, this approach is compared to a reverent cathedral building.

10. Intellectual Property

Opponents of OSS argue that open source threatens intellectual property rights. This is a specious argument designed to scare people away from learning more about open source. It's inspired by the most radical open source license: the GPL. Source code released under the GPL can't be included in closed, proprietary software. But any content created using open source software (including GPL software) still belongs to the author. For example, authors who create documents in OpenOffice still have copyrights on their work. With open source, only programmers can lose their intellectual property rights. They are willing to give up exclusive ownership of their code to contribute to the community and benefit from community-created programs. Authorship is still respected and rewarded.

Almost all software is created from existing software, using existing designs and common libraries of code. Hypothetically, if all software was licensed under the GPL then it would be almost impossible to create proprietary software without starting from scratch.

The technology offers many potential advantages and challenges, but it's alarmist to equate open source with the theft of intellectual property.

11. OSS in Ethiopia

Most users of third world countries like Ethiopia, who manage to buy hardware, do not want to buy, and more importantly, can’t really buy the software they need, because of the huge license fees. For example, the cost of Windows XP Professional is $269.99 and Microsoft Office Professional is $449.99[9] while the GDP per capital in the country is about $100.00 per year; this minimal cost is more than the annual income of an average Ethiopian. This encourages piracy of proprietary software, which has become a common practice.

Open source software are the most viable solution to the country not only because they are very cheap compared to proprietary software but also because they can be easily localized to Ethiopian languages and cultures. However, very few individuals and organizations in Ethiopia use OSS. Even those limited users of OSS are concentrated in international organizations. This is because of:

- Historical reasons: The first personal computers were introduced with Microsoft products and users do not want to migrate to new OSS products.
- Weak copyright enforcement: Users pirate proprietary (mainly Microsoft) products without penalty.
- Lack of compatibility: All OSS are not compatible with the existing propriety software.
- Shortage of support: There isn't enough OSS expertise in the country
- Adaptation of Microsoft products: Users are used to Microsoft products and do not want to learn a new way of working.

[9] http://www.amazon.com

However, this situation cannot continue forever. Very soon Ethiopia have to enforce copyright laws, it will be a member of WTO in few years. This will create a big chock, since users will have to pay for the software they are using, which is not affordable by most of them, or migrate to OSS in a very short time.

Therefore, it is preferable for computer users in Ethiopia to migrate to OSS, following their own calendar, before they obliged to migrate by external forces.

Using OSS software in Ethiopia will have enormous advantages:

- ✍ Legality: Illegal use of proprietary software has a negative effect for the country, in the long term, and should be discouraged. Currently, home users, organizations and companies are addicted to illegal use of proprietary software products. It will be more difficult to relieve the country from the dependency on pirated software where it will be obliged to enforce intellectual copyright laws.
- ✍ Beneficial for users: The cost of an Open Source Software is very cheap compare to proprietary software. Users do not pay for the license to use the software; they only pay for the service: modification and implementation.
- ✍ Income generation: It can develop local expertise and benefit from revenues that OSS generates (training, support, etc.)
- ✍ Lock-in prevention: Individuals and organizations that use OSS are free from the risk of lock-in to specific vendors of software product and services.
- ✍ Help to develop local ITC companies: OSS Lowers barriers to enter to ITC market for local developers and companies seeking to offer ICT solutions.

12. Common Open Source Software Licenses

There are many different software licenses in use. Various companies have created their own special license with sophisticated features representing their individual business model. Therefore it is very helpful to know which of them actually qualifies for open source or free software. Both the GNU project and the Open Source Initiative give some orientation in the license jungle

The following licenses are certified by the Open Source Initiative to conform the Open Source definition: GNU General Public License (GPL), GNU Library or `Lesser' Public License (LGPL), BSD license, MIT license, Artistic license, Mozilla Public License (MPL), Q Public License (QPL), IBM Public License, MITRE Collaborative Virtual Workspace License (CVW License), Ricoh Source Code Public License, Python license, zlib/libpng license. Among these Open Source Software licenses the most common OSS licenses are GPL, LGPL, MIT, QPL and BSD

1. **GNU General Public License (GPL)**

The GPL, created by the Free Software Foundation (FSF), is the most important open source license as most open source software is distributed under its terms.

The GPL permits unlimited free use, modification, and redistribution of software and its source code, but imposes three key restrictions on every licensee:

- If the licensee redistributes any code licensed under the GPL, it must guarantee availability of the code for the entire work for unlimited replication by anyone requesting it.
- If the licensee redistributes GPL code, it may not charge a licensing fee or royalty, but may charge only for distribution costs.
- If the licensee includes any GPL code in another program, the entire program becomes subject to the terms of the GPL.

The third restriction mentioned above often is referred to as a "viral" clause, because it causes GPL terms to apply to, or to "infect," software that incorporates or is derived from code distributed under the GPL, regardless of whether the program's developer intended that result or even knew of the presence of GPL code in the program. The motivation of this strategy is to prevent people from exploiting free software resource without paying (back) to the community by providing their achievements as free software, too. This is also used to preserve the source from commercial exploitation. The method is also called copyleft.

Violation of these restrictions may subject the offender to civil and criminal penalties for copyright infringement.

2. **GNU Lesser General Public License** (**LGPL**)

This license is more or less the same as the GPL except for one important feature: it permits linking with non-free modules. It was originally designed for standard libraries to speed up the adoption of free software since such licensed libraries provide an opportunity for proprietary software to run in a free software system. For this reason it is sometimes also called 'GNU Library General Public License'.

3. **MIT License**

The MIT license does not really restrict the software or its handling. The only condition is to include the copyright and permission notice in all copies.

4. Q Public License (QPL)

This is an open source license, which prohibits development of proprietary software based on the software licensed under the QPL. Anyone can make modifications and redistribute them in the form of patches along with original source code, as modifications must be distinct from the original. Generated binaries are allowed to have the same name as the original, which is important for dynamic libraries and similar components. Additionally, it forces the author of modification to grant the original producer the right to distribute the changes also under any other license, e.g. a proprietary one.

5. BSD License

The BSD License was developed by the University of California at Berkeley. It is very non-restrictive, it is similar to the GPL, but does not require derivative works to be subject to the same terms as the initial BSD License.
Under the BSD Licenses, distribution of source code is permitted, but not mandated for derivative works. Programs under the BSD Licenses can be combined with proprietary software.

13. Misconceptions about OSS

Some organizations and individuals avoid using of OSS, due to unnecessary fears of OSS.

1. **Is proprietary software fundamentally better supported than OSS? No**. There are actually two kinds of support for OSS: traditional paid-for support and informal community support. There are many organizations that provide traditional support for a fee; since these can be competed (an option not available for proprietary software), you can often get an excellent price for support.

   As an alternative, the user can get unpaid support from the general community of users and developers through newsgroups, mailing lists, web sites, and other electronic forums.

2. **Does proprietary software give you more legal rights than OSS**? **No**. Some have commented that "with OSS you give up your right to sue if things go wrong." The obvious retort is that essentially all proprietary software licenses also forbid lawsuits - so this isn't different at all! Anyone who thinks that they can sue Microsoft or other proprietary vendors when things go wrong is simply fooling themselves. In any case, most users aren't interested in suing vendors - they want working systems.

3. **Does OSS expose you to greater risk of abandonment? No**. Businesses go out of business, and individuals lose interest in products, in both the proprietary and OSS world. A major difference, however, is that all OSS programs are automatically in escrow - that is, if their original developer stops supporting the product, any person or group can step forward to support it instead. This has been repeatedly demonstrated in OSS.
4. **Is OSS economically viable? Yes**. There are companies that are making money on OSS, or using OSS to support their money-making activities.
5. **Will OSS destroy the software industry? Won't programmers starve if many programs become OSS? No**. It's certainly possible that many OSS products will eliminate their proprietary competition, but that's the nature of competition. If OSS approaches pose a significant threat to proprietary development approaches, then proprietary vendors must either find ways to compete or join the OSS movement.

One particularly interesting note is that the majority of software is not developed for sale. Organizations must pay developers to create it anyway. Thus, even if OSS eliminated all shrink-wrapped programs, it would only eliminate very little the existing software development jobs. And, since the OSS programs would be less expensive, other tasks could employ developers that are currently too expensive, so widespread OSS development would not harm the ability of developers to make a living.

OSS doesn't require that software developers work for free; many OSS products are developed or improved by employees (whose job is to do so) and/or by contract work (who contract to make specific improvements in OSS products). If an organization must have a new capability added to an OSS program, they must find someone to add it and generally that will mean paying a developer to develop the addition. The difference is that, in this model, the cost is paid for development of those specific changes to the software, and not for making copies of the software. Since copying bits is essentially a zero-cost operation today, this means that this model of payment more accurately reflects the actual costs. Hardware manufacturers also support the development of fee software

6. **If only OSS programs exist in a software category, will that completely eliminate competition? No**. Oddly enough, OSS programs sometimes compete with each other in a given functional area. For example, the text editors emacs (primarily GNU emacs) and vi (primarily vim) have dueled for decades. Sendmail is still a popular program for delivering email, but it has competition from other OSS programs such as Postfix and Exim. The desktop environments GNOME and KDE compete with each other, as do the OS kernels of Linux and the BSDs.
7. **Is OSS a "destroyer of intellectual property"? No**. You can use OSS products to develop private and proprietary information, and you can keep the information as confidential and proprietary as you want. What you can't do is use someone else's material in a way forbidden by law and this is true for all software, not just OSS.

   One interesting case is the "General Public License" (GPL), the most common OSS license. Software covered by the GPL can be modified, but any release of that modified software must include an offer for the source code under the same GPL license. Basically, the GPL creates a consortium; anyone can use the program, but you can't change the program or use its code in another program and make the results proprietary.

8. **Is there really a lot of OSS software? Yes**. Freshmeat.net counts over 21,000 software branches of OSS software as of October 2002. Sourceforge.net hosts 55,424 OSS projects all by itself (as of January 28, 2003).
9. **Is having the ability to view and change source code really valuable/important for many people? Yes**. It's certainly true that few people need direct access to source code; only developers or code reviewers

need the ability to access and change code. But not having access to how your computer is controlled is still a significant problem. Open source gives the user the benefit of control over the technology the user is investing in. The best analogy that illustrates this benefit is with the way we buy cars. Just ask the question, “Would you buy a car with the hood welded shut?” and the answer will be emphatic “No.” So ask the follow-up question, “What do you know about modern internal-combustion engines?” and the answer for most of us is, “Not much.”

We demand the ability to open the hood of our cars because it gives us, the consumer, control over the product we’ve bought and takes it away from the vendor. We can take the car back to the dealer; if he does a good job, doesn’t overcharge us and adds the features we need, we may keep taking it back to that dealer. But if he overcharges us, won’t fix the problem we are having or refuses to install that musical horn we always wanted -- well, there are many other car-repair companies that would be happy to have our business.

In the proprietary software business, the customer has no control over the technology he is building his business around. If his vendor overcharges him, refuses to fix the bug that causes his system to crash or chooses not to introduce the feature that the customer needs, the customer has no choice. This lack of control results in high cost, low reliability and lots of frustration.

10. **Is OSS really just an anti-Microsoft campaign? No. Certainly** there are people who support OSS who are also against Microsoft, but it’d be a mistake to view OSS as simply anti-Microsoft. Microsoft already uses OSS software in its own applications; Windows’ implementation of the basic Internet protocols (TCP/IP) was derived from OSS code, and its Office suite depends on the OSS compression library “zlib.” Microsoft could, at any time, release programs such as its operating systems as OSS, take an

existing OSS operating sytem and release it, or provide applications for OSS systems. There is no licensing agreement that prevents this.

11. **I've always assumed there's no free lunch; isn't there some catch?** If there is an OSS product that meets your needs, there really isn't a catch. Perhaps the only catch is misunderstanding the term "free." The GPL includes this (haiku) text: "When we speak of free software, we are referring to freedom, not price." i.e., OSS is not necessarily cost-free. In practice, it's still often a bargain.

    Naturally, if you want services besides the software itself (such as guaranteed support, training, and so on), you must pay for those things just like you would for proprietary software. If you want to affect the future direction of the software - especially if you must have the software changed in some way to fit it to your needs - then you must invest to create those specific modifications. Typically these investments involve hiring someone to make those changes, possibly sharing the cost with others who also need the change. Note that you only need to pay to change the software - you don't need to pay for permission to use the software, or a per-copy fee, only the actual cost of the changes.

6. Conclusions

OSS is often the most reliable software and in many cases has better performance and security, perhaps due to the possibility of worldwide view. OSS scales, both in problem size and project size. Total cost of ownership for OSS is often far less than proprietary software, especially as the number of platforms increases. Since OSS can be acquired with no or very little fee and they are easily localized to one's need, OSS is a viable solution to developing countries like Ethiopia

Annex III

Translations of OpenCms Workplace

One of the major works of the project is to customize OpenCms to Ethiopian languages. OpenCms has multi-lingual support, which enables developers to translate the workplace as well as on-line help system of the software. To fully localize OpenCms workplace 1177 vocabularies has to be translated. Among these about 225 are error messages.

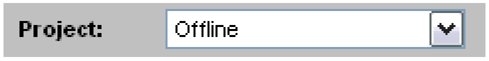
In this project the workplace of OpenCms is fully translated to Amharic. Among 1177 vocabularies of OpenCms workplace, I initially translated 600 of them.

To translate OpenCms workplace the following steps are required

a. Create a new module

To translate the workplace, a new module should be created. This is done as follows:

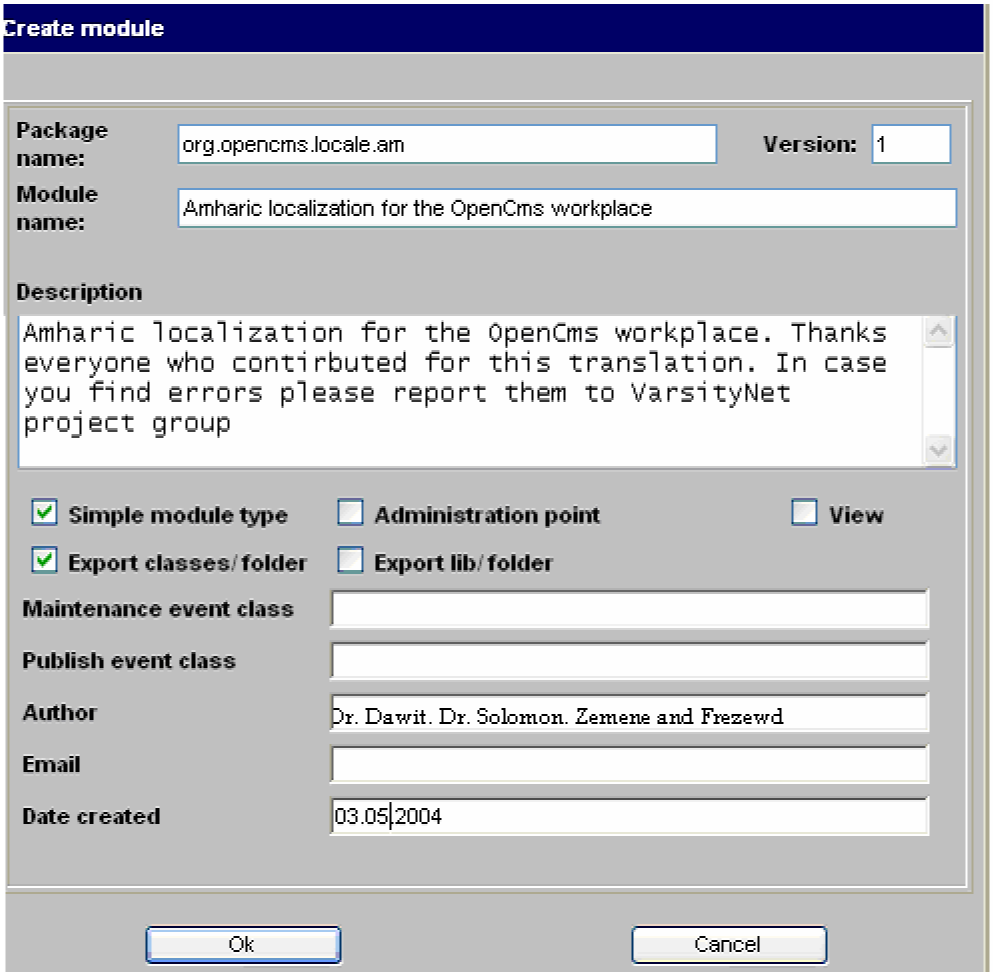
1. Go to the "Administration" view in OpenCms.
2. Make sure an "Offline" project is selected



Project: Offline

3. Click on the icon "Module management". a list of all installed modules will be displayed.

4. Click on the "Create module" button on the upper left of the screen. The "Create module" dialog opens. The dialog should be filled out as follows (in this example we used the "am" locale for the Amharic language):



Package name: org.opencms.locale.am

Version: 1

Module name: Amharic localization for the OpenCms workplace

Description

Amharic localization for the OpenCms workplace. Thanks everyone who contributed for this translation. In case you find errors please report them to VarsityNet project group

☑ Simple module type □ Administration point □ View

☑ Export classes/folder □ Export lib/folder

Maintenance event class

Publish event class

Author: Dr. Dawit. Dr. Solomon. Zemene and Frezewd

Email

Date created: 03.05.2004

Fig. 2. New module dialog box

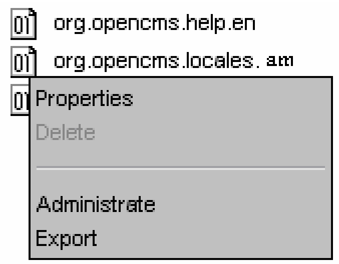
- Package name: org.opencms.locale.am (".am" for Amharic, ISO code for the Amharic language)
- Version: 1 (start with 1 and increase for new versions)
- Module name: Amharic localization for the OpenCms
- Description: This module contains the Amharic language localization files for the OpenCms workplace.
- Simple module type: checked (default)
- Administration point: NOT checked (default)
- View: NOT checked (default)
- Export classes/folder: checked (default)
- Export lib/folder: NOT checked
- Maintenance event class: [leave this empty] (default)

- Publish event class: [leave this empty] (default)
- Author: name of individuals who translate the workplace
- Email: contact E-mail address
- Date created: [dd.mm.yyyy] (creation date of the modulet)

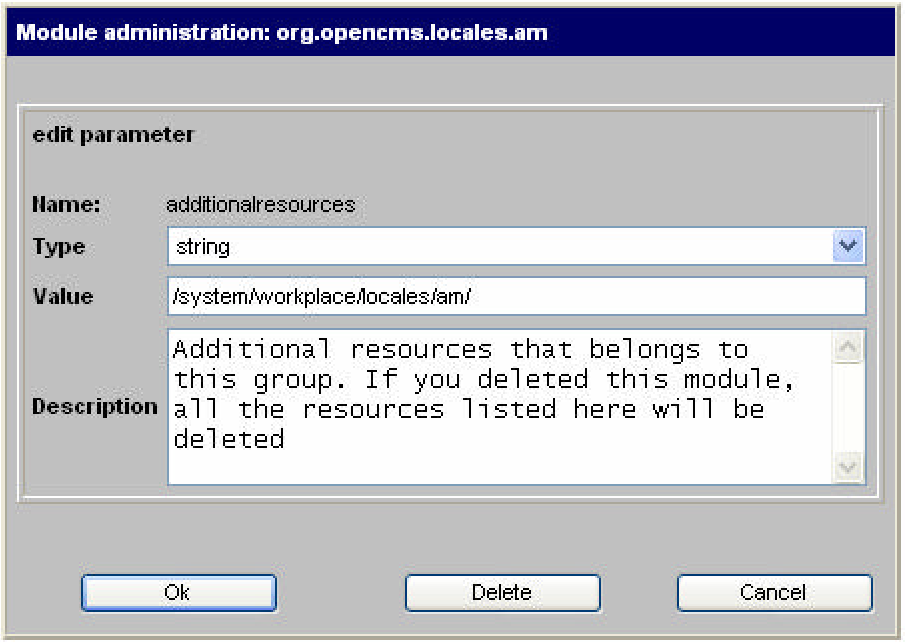
5. Click "ok".

The new module which should be displayed in the module overview page is created. Next step is to add a property to the newly created module.

1. Left-Click on the icon left to the new module package name. Choose "Administrate".



2. First the "Module administration" dialog will be displayed which looks in essence the same as the "Create module" dialog. To make any changes to the module data, it can be done here any time. Press the "Continue >>" button.
3. Now the "Module Dependencies" dialog will be displayed. Leave that as it is and just press "Continue >>". The localization module should not require a dependency.
4. The "Module properties" dialog will be displayed. Press "New" to create a new module parameter. The dialog must be filled our as follows:



Module administration: org.opencms.locales.am

edit parameter

Name: additionalresources

Type: string

Value: /system/workplace/locales/am/

Description: Additional resources that belongs to this group. If you deleted this module, all the resources listed here will be deleted

Fig. 3. Module administration dialog box

- Name: *additionalresources*
- Type: *string* (default)
- Value: */system/workplace/locales/am/*
- Description:*[can be left empty]*

5. Click on "Finish" button

The "Module properties" dialog should be displayed, and it should list the new entry for "additionalresources" as it is entered.

The "additionalresources" just tells the system to treat all resources at the given location and below as part of the module. So they are exported with the module in case you export it, which is important for the locales. Also, in case the module is deleted all resources listed at "additionalresources" will be deleted without a warning.

b. Adjusting of the basic folder structure

1. Select the view "Explorer" in the head of the OpenCms workplace. The current VFS tree of the OpenCms installation is shown. Open the /system/modules/ folder.
2. The folder named. org.opencms.locales.am should be displayed. Open this folder.
3. In the module folder the following subdirectories will be displayed: classes, default_bodies, elements and templates. Only need the classes folder for a localization module is needed. Delete all other folders in the module.
4. The classes folder will have a subdirectory structure org/opencms/locales/am/. Change the subdirectory structure to be com/opencms/workplace/ by renaming and/or deleting the existing directories. The folder name com/opencms/workplace/ must be the same for all localization modules, as this is the Java package name where the localization ResourceBundles are expected.
5. Now switch to the folder /system/workplace/locales/. Create a folder “am” with the 2-letter ISO code for Amharic language.

c. Creating the locale file

OpenCms 5.0 now uses standard java.util.ResourceBundles for the localizations.

Switch to the module folder of the new module and there to the classes/com/opencms/workplace/ subdirectory. In this directory, a file with the following name: workplace_am.properties should be created. Create a new file by clicking the "New" Button on the top screen and then select "Text" as file type. In the following dialog, input the name as explained above, "Title", "Keywords" or "Description" need not be filled.

For the creation / editing process of the new property the localization tool should be used. If some of the vocabularies are left out, OpenCms uses the default English labels for vocabularies not present in the localization file from the default workplace.properties file

The first three lines of Workplace_am.properties file contains the following entries,

```
version=$Revision: 1.1 $
name=Amharic
content-encoding=UTF-8
```

The version key value is generated by the CVS and can be left as it is.

The value of the key name is the language name as it will appear in the User settings dialog of the OpenCms workplace.

The value of the key content-encoding is the encoding that will be used for the workplace when switching to that language. The workplace actually has no encoding of it's own but will use the encoding set in the language file. "ISO-8859-1" can be used for all Western European languages but for Amharic it should be “UTF-8”.

d. Testing the new locale

First, all the changes should be published. Next step is a Tomcat restart. Unfortunately Java is unable to re- read ResourceBundles after a change, and so it must be started / stopped the Java VM every time changes are made to the locales if the changes in action are needed to be displayed.

After a restart of OpenCms log in and open the user preferences. The new locale you have just created should appear now at the "Startup options" tab. Select it to switch to your new locale.

Annex IV

Architecture of OpenCms

OpenCms is a client server application that can be used in HTTP-based environments such as the Internet.
OpenCms has three different layers

The presentation layer

The presentation layer consists of a web browser (Internet Explorer or Netscape Navigator) that is used to display and navigate through the HTML user interface.

The logic layer

The logic layer lies on the web server, which is extended by a runtime environment for Java servlets. The web server can be the Apache web server. All Java classes are installed in the servlet environment on the server. The OpenCms servlet provides the interface to the presentation layer as well as the interface to the database layer. It uses the HTTP protocol to establish the communication between the client and the application on the server (in this case OpenCms). The database is accessed through the JDBC interface.

The data layer

The database contains tables for resource, user and property data.

- ✍ CmsObject: All resources are accessed via the CmsObject. This is the interface to the OpenCms system for the Module Developer. The CmsObject is initialized with the user data.
- ✍ Resource Broker: The resource broker checks requests for resources and performs them if the user has the appropriate access permissions. It’s configuration depends on the database that is used.

✍ Launcher Manager: The Launcher Manager determines the launcher that will generate the output with the requested content based on the type and/or content of the selected file. If the template launcher has been selected, the Template Mechanism is started.

✍ Template Mechanism: The Template Mechanism creates HTML pages based on the content in its structured form. Content definitions enable to access content that originates from sources (e.g. Virtual File System (VFS), database, file system).

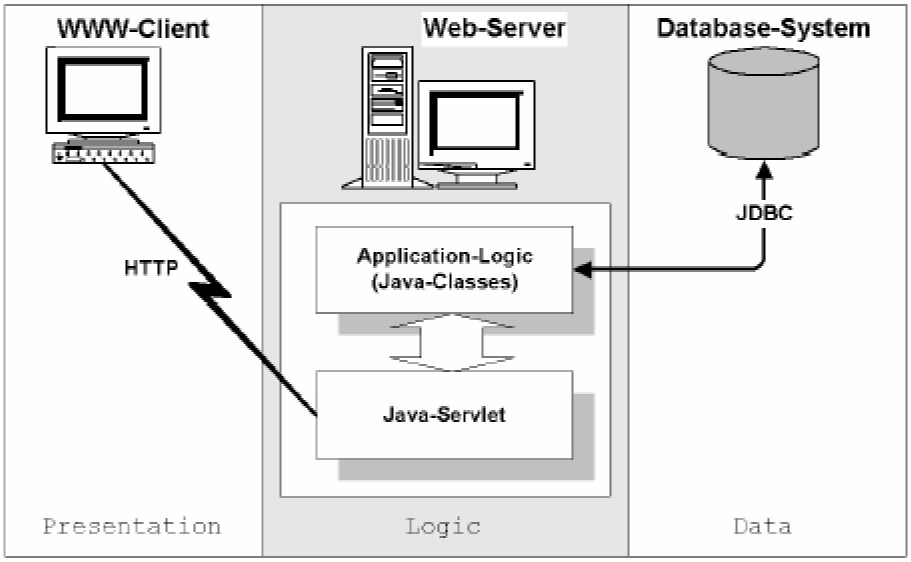


Fig. 1. Architecture of OpenCms (source: http://www.opencms.org)

Annex V

Installing OpenCms

To install OpenCms (Version 5.0)t he following steps should be completed.

a. Install the Java JDK

Install Java JDK 1.4 or later.

b. Install Tomcat

OpenCms 5.0 requires a Servlet 2.3 / JSP 1.2 standards compliant container. Tomcat 4 is the reference implementation of this Standard. Install Tomcat 4.0.x and Tomcat 4.1.x. into a folder of your choice. This is the CATALINA_HOME folder.

Set the environment variables CATALINA_HOME and JAVA_HOME.

To make sure Tomcat works with the correct charset to read and write, set the environment variable CATALINA_OPTS to -Dfile.encoding=UTF-8.

c. Install MySQL

Install MySQL, on Windows-based systems MySQL has to be installed on the C:\ drive and should be registered as service using /mysql/bin/mysqld -install.

Start the MySQL server by running the service (WIN32) or executing (your MySQL path)/mysql/bin/mysqld (UNIX).

d. Deploy the opencms.war file

The opencms.war file should be copied from the binary distribution ZIP file to CATALINA_HOME/webapps/. Start Tomcat. Tomcat will now deploy the web application OpenCms.

e. Install OpenCms using the Setup-Wizard

Start the Setup-Wizard by pointing the web browser to http://localhost:8080/opencms/ocsetup. The port 8080 is only used to start Tomcat in standalone mode.

f. Use the system

To use the system, login with username: Admin and password: admin. This login name and password should be change soon. The login URL of OpenCms in a default configuration is: http://localhost:8080/opencms/opencms/system/login/

Annex VI

Interview of government offices to identify needs of Multi-lingual E-government on-line document management platform development

Interviewers: Dr.Dawit Bekele, Ato Frezewd Lemma and Ato Zemene Adgo

Interviewee: Ato Gemechu Geleta

Date of Interview: October 24, 2003

Time of Interview: 2:00PM - 3:00PM

Location of Interview: At Ethiopian Science and Technology Commission, Deputy Director's Office

Type of Interview: Face-to-Face

The semi-structured interview conducted with deputy director of National Computer and Information Center, Ato Gemechu Geleta, includes the following questions.

Question 1: What are the major objectives of the Information Communication Capacity Building Programme?

Question 2: What are the major objectives of the project that will connect all Woredas of the country (WoredaNet)?

Question 3: What are the main problems of the currently used method of document exchange and management?

Question 4: What are the major reasons that justify such a big investment?

Question 5: What types of applications are being developed/planned to exchange documents using this infrastructure (the woredanet infrastructure)?

Question 6: What kinds of content do you expect (e.g. language, etc)?

Question 7: What organizations/users are the main focuses of the project?

Question 8: Is there any pilot project?

Question 9: Have you thought about the architecture of your system (Distributed/ Centralized)?

Question 10: Have you thought about security, privacy, etc?

Question 11: Have you thought of the consequence of having a low bandwidth and unreliable connection?

Response to question# 1:

The major objectives of Information Communication Technology Capacity Building Programme are two fold:

- Since there is no ICT capacity in this country, the first aim of the program is to build the ICT capacity it self. This includes human resource (ICT professionals), ICT infrastructure and content. The government is spending big amount of money to establish ICT infrastructure and content development in the country.

- Since ICT has cross-cutting effect across the different development sectors, the second objective is to use ICT as an instrument to facilitate the Economic development of the country by supporting other sectors like Agriculture, Health, and Education; these have been given higher priority by the government of Ethiopia.

Response to question# 2:

Information Communication Technology Capacity Building programme has six components

- ICT for good governance
- ICT for other sectoral development
- ICT for private sector development
- ICT for policy, regulatory area and etc

- ICT for community access and
- ---

The major objective of WoredaNet project is for good governance. It has two phases:

Phase 1: Video Conferencing

The infrastructure will connect all the Woredas of the county and the federal government. To conduct or participate in a conference with the federal government, professionals or other officials working in different Woredas won't go to Addis Ababa or other cities. They can conduct or participate in the conference virtually, without going to the place where the conference is held. Some of the benefits of using videoconferencing are:

- Saves travel time and expenses
- Improves use of executive time.
- Speeds up decision-making.
- Keeps meetings brief and more focused than face-to-face
- Enables top management to quickly and effectively communicate with employees sitting in multiple locations.
- Allows virtual project management via video and data conferencing with geographically dispersed peer groups at short notice.
- Provides an effective way of delivering cost-efficient training to individuals found in Woredas without the requirement to consistently travel to central locations.
- Creates a medium for conducting interviews.
- Keep meetings brief and more focused than face-to-face

Phase 2: Content development

This is E-government strategy; this includes language translation and document exchange and management using multiple languages. Under the development of content two organizations, UNDP & Microsoft are involved. The first one is studying what will be the structure of the e-government and the latter one is developing some portal services especially for finance, pension, and parlama.

Response to question# 3:

Till today, documents are exchanged between regions and the federal governments or between one Woreda and another Woreda manually, i.e. through post office or using cars or motorcycles. This method of document exchange is unreliable, unsecured and above all very slow. That is why the woredanet, which includes the VSAT infrastructure needs to be put in place and the first phase, video conferencing, and the second phase, document management, are considered to be complementary.

To solve the problem of document exchange there will be a data centers at the woreda levels. The data center will expand with sectors.

Response to question# 4:

First of all, decentralization, which is the political strategy of the Ethiopian government, is information intensive and we all know that information is power. To centralize the decentralized ones and to strengthen Woredas of the country fast and easy information sharing and communication is very essential. Professionals are now in different Woredas of the country; to maintain and attract these professionals they should have access to the Information. To conduct videoconference, distance learning and exchange documents and information the infrastructure should be first established

Response to question# 5:

In addition to videoconferencing and content development, other 16-17 applications have been identified and are waiting to be developed. Some of these are:

✍ Financial management

- Human resource development
- Procurement management
- Taxation system
- Land use management
- Resource management
- Tourism development, etc.

Response to question# 6:

Different Woredas of the country speak different languages. English language is not used frequently in different Woredas of the country. Therefore, language translation among the local languages is very important. In addition, document management and exchange using multiple languages is very crucial. Because these help to protect and respect the different cultures in the country.

Response to question# 7:

Private sectors, non-governmental organizations (NGO) and communities at the woreds level are the main focuses of the project.

Response to question# 8:

Yes. Microsoft Company has already started pilot projects in Financial sector and social security. The main focus of these pilot projects is to exchange documents and information from Woredas to federal government. The pilot project is being implemented in Oromia region around Deberezeit.

Response to question# 9:

The data center architecture will be clarified by Ato Zelalem.

Response to question# 10:

I haven't thought of security architecture and privacy issues. Even though they are essential, they are not considered yet.

Response to question# 11:

The infrastructure will have a bandwidth that is expected to support videoconferencing and document exchange without any problem for three to five years. Therefore, the consequence of having low bandwidth and unreliable connection is not considere d.

Summary of the interview:

The government of Ethiopia is investing such a big money to build the capacity of Information Communication Technology. The main objective of ICT Capacity Building Programme is to get the most out of it by facilitating the Economic development of the country through supporting prioritized sectors like Agriculture, Health and Education.

One of the projects of ICT Capacity Building programme of the government is WoredaNet which will connect all Woredas of the country each other and to the federal government. The main objective of WoredaNet is to establish infrastructure through the country. Video conferencing is the first phase of the project; the second phase is content development, which includes language translation and document exchange and management using different languages.

The currently used manual method of document exchange and management that different government offices use is unsecured, unreliable and more over time consuming.

The project is my original work, has not been presented for a degree in any other university and that all sources of material used for the project work have been duly acknowledged

Zemene Adgo _________

Dr. Dawit Bekele (Advisor) _____________