Acknowledgements

Without the help and ingenuity of several people, this research project could not have been done.

First and foremost, I am pleased to acknowledge the consistent support of my advisor Dr. Dawit Bekele, whose valuable guidance and advice made this work to come to an end. Without his management capability and understanding of the jobs, I could not finish this project within its time limit. Thank you very much for all of your polite way of handling things.

I have a great appreciation and admiration to Daniel Yacob who gave his time for the advancement of Ethiopic software. He gave me a c-and perl code, which he wrote for an online Ethiopian calendar so that I can use it in my project. When I faced different problems during the project time, he was the second person to contact.

I am very much grateful for Dr. Solomon Atnafu who took his most valuable time to check the 600 translations I made and did many important corrections.

I want to give my thanks to my brother Shewatatek Lemma and other RVC AA campus staff who covered my administration responsibilities in the college when I was busy with the project.

My acknowledgements also goes to Menasse Zewdou from sun Microsystems, for his help in finding solutions for different problems that I faced at different times.

Last but not least, my thanks goes to also to the three undergraduate students Tilaye Yismaw, Taddese Kebede, & Abinet Ayalew who helped the project during their semester break.

Abstract

Content management systems, which take advantage of the client-server communication of the Internet and intranet, provide a great opportunity to manage an organization documents and contents. Content management systems software exist both as proprietary and open source.

Open source software are getting more attention around the world and it should be the same for Ethiopia too. This is because open source software can be acquired with minimum total cost of ownership and can be localized to support Ethiopian language and culture without the permission of the owners.

This project started with by conducting Content Management System requirements of government organizations and identifying development strategies. Then it has been continued by performing the following major works of the project.

1. The revision of related works which included topics that are necessary for the building of an open source content management system for Ethiopia.
2. The testing and benchmark study done on Zope and OpenCms content management system software in order to choose one of them.
3. The development of Gregorian to Ethiopic date and time conversion program and integrating it in OpenCms.
4. The translation of half of the OpenCms user visible words/phrases and preparing the system environment.
5. Localization of OpenCms so that it can work with Ethiopian culture.
6. Extending OpenCms to incorporate a stand-alone Ethiopian calendar module so that it is available on the Internet.

Table of Contents

List of Tables..page

List of Figures

List of Annexes

Acronyms and Definitions

Validation – The process of evaluating software at the end of the software development process to ensure compliance to software requirements.

Verification – The process of determining whether or not products of a given phase of the software development process fulfill the requirements established during the previous phase.

VFS- Virtual File System

Internationalization: "Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalization takes places at the level of program design and document development."

Localization:"Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold."

Locale: The features of the user's environment that are dependent on language, country, and cultural conventions. The locale determine convents such as sort order; keyboard layout; date, time, number and currency formats. [*11*]

Unicode: is a universal character set, i.e. a standard that defines, in one place, all the characters needed for writing the majority of living languages in use on computers

Character set: the set of characters one might use for a particular purpose

OSI: Open Source Initiative

1. Introduction

“It is likely that the information revolution widens the gap between the poor and the rich by creating yet another divide, the information divide”[4]. One area, which facilitates this, is a technology that helps to properly manage information in an organization.

In the last decade electronic document and content management systems have become crucial for private, government, and non-government organizations since they are producing increasing number of electronic documents. 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]

Content management systems enable organizations to effectively and efficiently administer, handle, maintain, deploy and deliver online content using their Intranet.

The need for greater efficiency in handling documents fueled the rapid development of document and content management software systems. There are much software that is developed in the last decades that can be used to manage documents as well as their contents. These software are both open source and proprietary.

In Ethiopia, most organizations are not benefited from document/content management software. Some of the reasons are lack of complete knowledge of open source software, lack of awareness and knowledge of content management systems, price of proprietary content management software, lack of localized software, which use the language and culture of the average citizens.

The main objective of this project is to develop an open source content management system localized to the Ethiopian context and that will help individuals, private, public, governmental, and non-governmental organizations to better manage their electronic content. For this purposes an open source content management system has been localized and customized to include Ethiopian specificities such as the language, the calendar, measurement unit, currency symbol, holidays etc.

This report presents the problem statement of the project, the methodology used to achieve the target, related works done by others, and works done in this part of the varsity net project.

This work is done in coordination with another work by Zemene Adgo that handles some of the problems of the project. Both works were supported by AAU VarsityNet, a project of Addis Ababa University, UNECA and Ford foundations.

2. **Problem Statement**

This project will analyze the requirements of the Ethiopian government organizations in content management software and develop a system that responds to these requirements.

The requirement analysis will focus on Ethiopian local governments (Woredas) specificities since other general requirements of a content management system are well known. The target system on which the system is to be used is the woredaNet infrastructure under development.

Several strategies will be studied for the development acquisition of the system including developing the software from scratch, customizing existing proprietary or open source content management software.

Major emphasis will be given for the choice of the development strategy to produce the software that responds to the requirements within the limited time and budget.

Since a major objective of this project is to develop a content management system that is localized to the Ethiopian context, this will focus on identifying the localization requirements as well as on identifying the best strategy for the localization and implementing this strategy.

3. Methodology

The following methodologies have been used for the requirement analysis and the development of the system.

For the requirement analysis methodologies used are:

- Interviews: Formal and informal interviews were conducted with government officials to find the content management needs of government offices especially woreda offices.

- Reviewing of previous studies: Documents that presents the content management needs of government organizations were reviewed.

- Literature review of content management system: literature review was done to find out typical functionalities of existing in content management systems.

For the development of the system two strategies were proposed:

- The first strategy was to develop a content management system from scratch.
- The second strategy was to customize/localize existing proprietary or open source software.

To choose between the two strategies the pros and cons of each strategy has been studied. Based on the study the second strategy was selected. Consequently, several different content management software have been studied and the one that satisfied most of the requirements and easy for customization has been selected.
Generally for the development of the system the following procedures have been followed:

- Reviewing available open source content managmenet system
- Getting experience from experts
- Installing open source content management software
- Testing of open source content management software
- Selecting an open source content management software
- Developing necessary programs for localization
- Localizing Open Source Content management software
- Translating the vocabulary of the selected open source content management software
- Testing of the prototype
- Writing a project report

4. Literature Review

This review of related works focuses on open source software, open source content management system software, localization of software and works that help to localize software for Ethiopian culture and languages.

4.1 Open source software

Much work has been done around the concept of open source software by different organization and individuals. Reviewing the concept of open source software was necessary due to this project objective. As indicated in the introduction section the project objective is to develop an open source content management system. As a result it was important to know what the meaning of open source is, and what are the things to understand in order to use them. Consequently, the review in this section focuses on the definition given by OSI.

Generically, *open source* refers to a program in which the source code is available to the general public for use and/or modification from its original design free of charge, i.e. open.[18]

The open source initiative (OSI), which is the deriving force behind open source software, put the definition of open source based on the distribution terms that must be applied to any open source software.

Accordingly, 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 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.

- **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.

- **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.

- **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.

- **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. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

- **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. 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.

- **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.

- **License Must Be Technology-Neutral**

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

4.2 Content Management systems

One type of area, where open source software are matured, are content management systems (CMS), which are the target of this project. As a result they are reviewed to see what features do they have in general and to what extent they satisfy the requirements of the project.

Before CMS's came to the stage, it was document management systems (DMS) that play important roles in an organization. The difference between these two type of software is that document management systems are deal with the entire document as a whole where as content management systems deal with not only the document in its entirety but also the content of the document too.

The transformation of DMS to CMS happened when the time of Internet bloom in the middle of the 90's. At this time document management systems vendor reorganize them selves as content management systems.

Currently there are more than 80 open source content management software. Most of which have the following common features:

Feature	**Description**
Document repository	This is the main purpose of any content management system. Most CMS's use databases as their repository. Others use flat files or XML files.

Multilanguage support	This is the functionality from cms's to make a site to work with many languages.
Searching	Functionality used to find documents by their content, and index fields such as date, file name, file type, and size.
Workflow	Workflow is an IT technology, which uses electronic systems to manage and monitor business processes.
Security	Security is about, user permission and rights, supporting LDAP, supporting SSL, plug in authentication, login history, session management and so on.
Versioning	This helps to store a previous version of a changed document.
Localization	The feature used to adapt the software to ones local condition.
Browser-enabled work environment	The feature used to create, update, and manage any page element through the generic web browser.
Template mechanism	Used to present page elements to the user of the web page.

Web site creation and maintenance	In content management system there are many features used to create and maintain a web site. Some of the common features include are poll, survey, authoring tools, discussion forum, link management, creating document for multiple targets, graphics and multimedia management, personalization, and templating
Plugin API	This is the feature used to expand the system in order to incorporate a new functionality in to the software.

Table1. Features of Content Management systems

All the above features are in agreement with the features required in the software to be developed by this project.

Although most CMSs have common features, each one of them has its own distinctive feature that separates it from others. When we come to localization and customisations, some of them have a high learning curve.

To classify open source content management software, it is possible to use their general features as well as the platforms they need, by the language they are written and can be extended by. When we say platforms, it includes the operating system, the database, and the web server on which a given CMS run.

Most open source content management system uses open source platforms. For example, many of them can use Linux as their operating system, MySql as their database, Apache as their web server. Beside these, some of them like Zope, use their own database and web server. That means without having any additional software they can be used as a complete content management system out of the box.

4.3 Localization

Localization is the main topic of this project. It was necessary to know what localization is, and what are the procedures to follow in order to localize a given software. In this section the definition of localization, its relationship with internationalisation, the importance of Unicode in localization and resources necessary for Ethiopian localization are presented.

A lot of content management system provides a way to translate to other languages. The process of translating software to other language and customizing it to support other cultures and convention is called Localization.

A program that can be localized to other language and culture is called an internationalized program. Software can be divided in to four by their degree of internationalization:

1. **No international support**: The application works in one language. If that language is not English, it probably works on only specific language versions of Windows.
2. **Locale-dependent source**: Different code base must be written and maintained for different cultures.
3. **Single-source, locale-dependent binary**: A single code base is written but separate compilations must be made for different languages or different Windows versions.
4. **Single-source, single-binary**: A single code base and single compilation satisfies all language and platform versions. [10]

If a given software has the fourth degree of internationalization, it has the following structure:

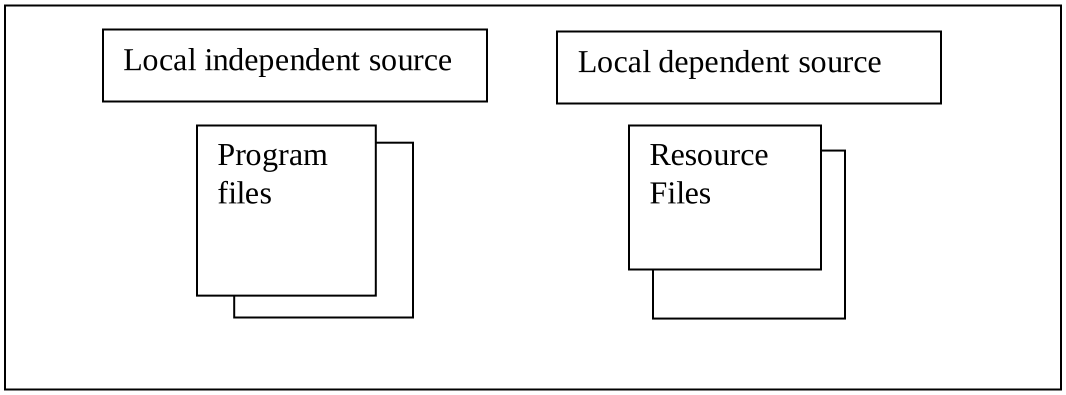


Figure 1. Structure of fourth degree internationalised program

Localization of software is needed due to the cultural and regional differences in the following components of an application:

Date format, time format, time zone, calendar, number format and digits, currency, collation, character set, case rules, hyphenation, justification, quote character, title & address & telephone format, user interface language, writing direction, list separator, license plate format, form layout (paper size), units of measurement (pounds, grams, stones, inches, meters, etc.), keyboard, input method, Colors, and Graphics.

For example, in the US, the color red often implies danger. However, in China, red connotes prosperity. It is possible to display warning messages in red for a web application that targets a US locale; in China, one would choose a meaningful color for that culture.

Many localization works showed, localization involves adjusting one or more of the above differences to make a given software work in the intended locale.

Again in many literatures it is showed that an internationalised program, which uses Unicode, is the simplest to localize to other language, culture and conventions. The following statement proves this idea.

"A Unicode encoding can support many languages and can accommodate pages and forms in any mixture of those languages. Its use also eliminates the need for server-side logic to individually determine the character encoding for each page served or each incoming form submission. This significantly reduces the complexity of dealing with a multilingual site or application." [9]

For example an application that is fully browser based, which is the case for most content management systems, the browser that they use must support a Unicode so that they can be used with other character sets than Latin set.

In case of CMSs, beside the need for the browser supporting of Unicode, it is also important that HTML, XHTML, and Java editors support Unicode in order to localize these browser-based applications. In cases where these editors do not support Unicode but if the browser supports Unicode, it is possible to insert Unicode characters by using Numeric Character References (NCR), to represent any Unicode character using only ASCII characters. For example, the following are different ways of representing the character ሀ:

&#x1200;

A hexadecimal NCR. NCRs are a type of escape. All NCRs begin with &# and end with (;). The x indicates that what follows is a hexadecimal number representing the scalar value of a Unicode character, i.e. the number assigned in the Unicode code charts.

&#4608;

A decimal NCR. This uses a decimal number to represent the same scalar value.

One point worth special note is that values of numeric character references (such as &#4608; and &#x1200; for ሀ) are interpreted as Unicode characters - no matter what encoding used in a document.

Beside Unicode support of applications, fonts are another important things to have during translation. For Ethiopic localization a good source of Unicode fonts and keyboard exist at the "The Ethiopic Unicode Resource Page" which website address is :

http://www.abyssiniacybergateway.net/fidel/unicode/

At this site the following fonts and keyboards can be downloaded freely:

Keyboards

UniGeez

Keyman 5.0 for Amharic and Tigrigna

Fonts

TrueType

Code 2000: http://home.att.net/~jameskass/CODE2000.ZIP

Ethiopia Jiret: http://www.senamirmir.com/download/jiret.zip

GF Zemen: ftp://ftp.ethiopic.org/pub/fonts/TrueType/gfzemenu.ttf

Titus Cyberbit: http://titus.fkidg1.uni-frankfurt.de/unicode/tituut.asp

Visual Ge'ez: http://www.custorc.com/downloads/VGNSetup.EXE

Concerning localization works done for Ethiopia, there is only one work that is available to quote. It is the localization of Linux done by Mr. Daniel Yacob. In this work Daniel has made localization at the different levels of linux operating system and translated more than 5000 english words/phrases used in linux to Amharic words/phrases. Since there is no a formal publication of this work, it was not possible to review the work in its entirety.

5. The project works

This section presents a report on the works done in this project. The works done in this project are: selecting development strategies, testing zope to see whether its features satisfies the requirements of the project, a benchmark study done to select between zope and OpenCms, identifying the kind of localization requirements exist in OpenCms, Localization of date and time and adding a module for Ethiopian Calendar.

In addition to these main works, the report involves report on validation and verification methods used in the development process, different lessons learned out of this project, and works needed to be done in the future in order to further develop the system.

5.1. Selecting a strategy

As described in the methodology section, the two proposed ways of development were developing from scratch and customizing already existing software. To decide between the two alternatives, it was necessary to study the pros and cons of the two. Comparing the requirements of the project with what others did to achieve such requirements, it was possible to saw that developing content management software from scratch within the project deadline is invisible.

The other alternative, customizing and localizing from exiting CMSs, was selected to be the development paradigm. Accordingly, testing of the selected software (Zope – by the author of this document and OpenCms-by Zemene Adgo) were performed.

5.2 Testing different features of the zope software and the report about the test

In this project, the first two works were revising open source software in general (principle of open source, licenses etc) and open source content management systems (CMS) software in particular (see Annexe B). In the latter, a lot of open source CMSs were studied and out of them 8 were selected, which provides almost all the necessary features required by the project (see Annexe B). The most important criteria used to select the 8 software were:

- Whether the software is supporting multiple languages or not.
- The support type they have: do they have commercial support, mailing list, user friendly community, etc
- Their system requirement: This is a criteria that looks in to whether the software require a commercial software or not? If it requires, it wouldn't be selected
- The features they have: whether the software is rich with its features or not?

- Content management framework: Whether they have the framework to develop a new CMS based on them or not?

Using the above criteria the software selected were: Typo3, EZ publish, Bricolage, OpenCms, Midgard, Webgui, Zope, and Twiki.

Being these 8 content management systems software presented for a group of experts, three of them were passed for a final benchmark study. These chosen software are WebGui, Zope and OpenCms. For zope, the group recommended using Plone, which is a ready-made content management system, based on Zope.

At the beginning we dropped WEBGUI due to lack of documentation and instead of it we have added Typo3. We tested TYPO3 together with the other group, which is assigned to test OpenCms;but its learning curve seems high and we dropped it too.

Out of the three software the testing of Zope was assigned to be included in the project of localizing open source CMS for Ethiopia. Accordingly, Zope was tested for the following features, and the result of the test is presented in the following table:

Feature	Plone-Zope	Remak
Localization	Yes	Zope has a resource file that contains all the user visible texts. It is a matter of translating this file to localize zope for other languages. For other localization things there was a shortage of documentation.
Customization	Yes	In plone customization is achieved by writing scripts using DTML, python, or ZPT.
Workflow	Yes	In Zope workflow is defined using states, transitions, variables, wordlists, and permission. The zope workflow is one of the strengths of zope.
Versioning	Yes	In Zope, all modifications are archived and it is possible to undo to the previous versions. Using a version object, it is possible to work on many versions of a given object.
Email notification	Yes	Zope can send e-mail notification to preferred users or user groups from the workflow system.
Forum	Yes	Zope has a simple method to add discussion forums in any page.
Download/Upload documents	Yes	Upload and download are simple in zope
Scheduling	Yes	Scheduling using Zope is very simple.
Linking	Yes	Zope have simple method of linking related documents/pages
Check in/ check out	Yes	In Zope this is done internally.
Auditing	Not discovered	Zope keeps track of information who accessed a page when and type of modification on the page.
Help system	Yes	Zope have online help system
Searching	Yes	Zope support search engine.
Security	Yes	Plone-Zope has different security mechanism.
Copy, move, delete, and browse	Yes	Both Plone-Zope have simple method of copying, moving, browsing and deleting objects.
Langage	Python, C/XML	
WYSIWYG		Achieved by adding WYSIWYG editor like Dream Weaver

Table 2. Zope Test result

After finishing testing Zope for the features it supports, the easiness of accomplishing different task was compared to its competitor- OpenCms through a benchmark study.

5.3 How the benchmark study was performed?

For the benchmark study three 4th year under graduate students were selected and they were given all the requirements of the project. After downloading and installing the three software on three computers, each student tried to see that whether his software is easy to use or not. After collecting a report from the students, a practical demonstration was made by them. From the report and practical demonstration a final report were produced.

The result of the benchmark study showed that both zope and opencms fulfills the requirements and both of them are not so difficult to use, even if there are differences in the way they accomplish the same tasks. Because of this equality of the two software, the group determines to use the language by which these two software can be customized and extended.

Zope is written and can be customized by phython and its own scripting languages DHTML and ZPT. Both of which are not totally known by any of the team member; where as OpenCms is based on Java, Java script and xml. All of the team members have experience on java, java script and xml. As a result, with this single criteria OpenCms was selected. If that had not been the case, since the learning curve of phython, Dhtml and ZPT, will be high, the project would not meet its deadline.

5.4 Localization Requirements

Once OpenCms is selected, the next step was to see what kinds of localization and customization is necessary to make OpenCms work with the Ethiopian culture and conventions.

Consequently, the following requirements were collected after a close study of OpenCms for both cultural and technological aspects of localization.

Under the cultural aspects, OpenCms was studied whether it has the following features or not:

- Date, time and number format
- Measurement units
- Idioms
- Pictures
- Sounds
- Fashion
- Religion
- Values and symbols
- History
- Laws
- Colours
- Ethics and morality

After this study it was found that OpenCms uses Gregorian date counting system. Other cultural aspects are non-existent. As a result the following functional requirements found to be required from the customized Opencms:

- Date and time: Since OpenCms works with Gregorian system, it should use Ethiopian date and time counting system whenever Amharic is selected as a user preference language. The system should differentiate between long and short date formats and displays the year, month and day according to the locale convention.
- Calendar usage: The system should use an Ethiopian calendar, whenever the language used is Amharic and a user displays a calendar.

Under the technical aspects, OpenCms was studied for the following things:

- Character encoding
- Fonts
- Indexing, sorting and searching of data
- Rearrangement of shapes, images, texts

It is found that OpenCms uses ISO-8859-1 as its default character encoding. Since this ecnoding is for latin character set, it could not be used for Ethiopic character set. As a result, it was necessary to customize OpenCms so that it supports Unicode character set like UTF-8. Once it is customized to use UTF-8, it is possible to use Unicode fonts.

For the indexing, sorting, searchin texts, and rearrangement of shapes, images, and texts no requirements are found.

5.5 Customization of OpenCms

To use a calendar other than the Gregorian and localize the date and time formats in OpenCms, modifying the logic of the program was the only way. This is because OpenCms does not provide a single place to localize date and time format as well as inserting a new calendar into it as it does for the language part.

The following architecture of OpenCms shows its components and indicates what to understand in order to modify it.

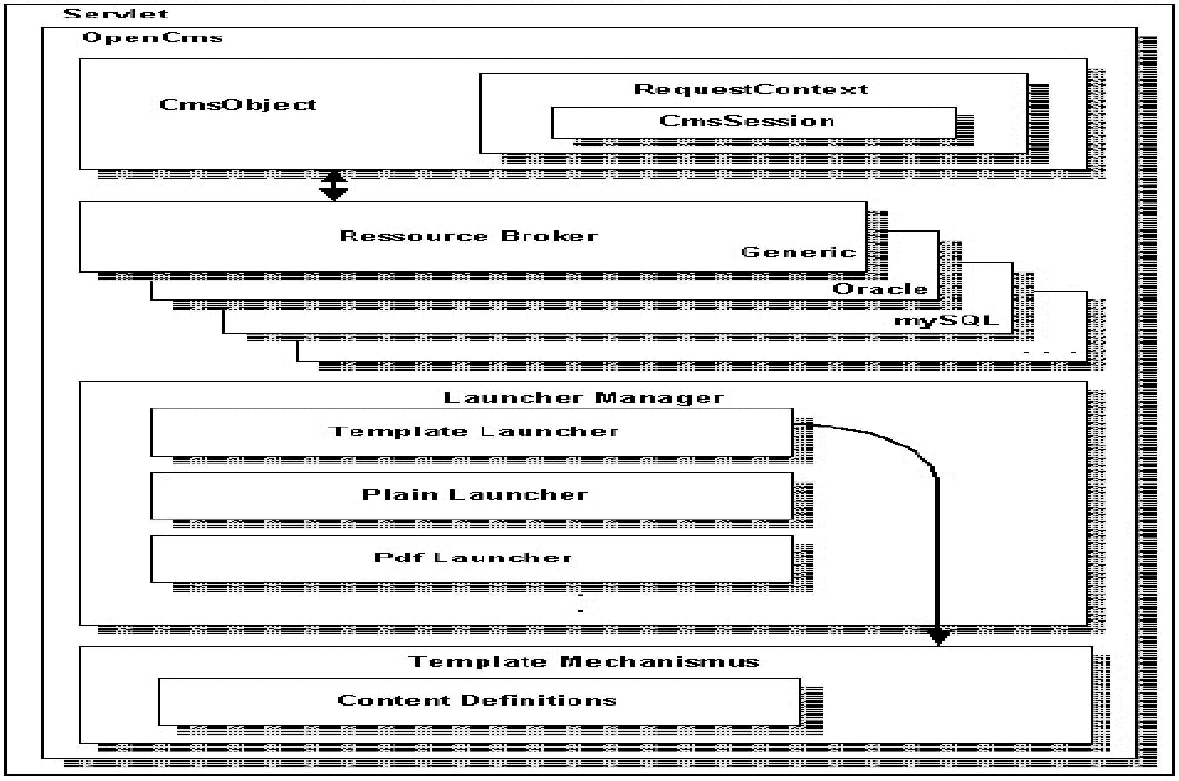


Figure2. Architecture of OpenCms

Out of these components understanding the OpenCms template mechanism was the most important one.

5.5.1 Date and Time localization

The purpose of date and time localization is to make OpenCms aware of Ethiopian date and time system.

It is not simple to customize OpenCms for ones culture of date and time usage when compared to translating OpenCms to another language. This is because date and time are hard coded in the application i.e. there is no single place to give ones date and time format, number of months, current year and so on. So it is necessary to modify the code, and then to build (compile) OpenCms from the modified source code.

After studying the code it became clear that it is enough to display Ethiopic dates when Amharic language is selected i.e. no need of storing Ethiopic dates and time. It simply needs to convert the stored date and time to Ethiopic date and time when date and time information are being displayed for the user.

To convert Gregorian to Ethiopian date and time, inserting new programs into OpenCms were needed. For this, two Java classes, Ethiopic.java and Day.java, were developed (See Annexe D) and incorporated into OpenCms.

Incorporating the two new classes in to OpenCms was a two-part task. The first part is to study the 10 packages of OpenCms to put the new classes at the right package. In average each package contains 38 java programs.

At the beginning it was not straightforward to get the right place in the situation where there is no enough documentation. But lastly it was possible to see by the trial and error study of the packages, the boot package is the right place to store the new classes in this date and time localization case.

Now we have the date converters integrated into OpenCms. The second part is to make all the necessary modifications in original OpenCms classes that display dates to the user. The aim of this modification is to use the converter programs before the dates are displayed for the user. The following table presents the summary of the modifications made in this regard.

No.	**Class/package Affected**	**Method modified and classes imported**
1	Boot package	Ethiopic.java and Day.java are put in this package
2	CmsAdminModuleAdmin.java	private void fillHashtable(CmsObject cms, I_CmsRegistry reg, Hashtable table, String module)
3	CmsModulelist.java	public Object handleSpecialWorkplaceTag (CmsObject cms, Element n, A_CmsXmlContent doc, Object callingObject, Hashtable parameters, CmsXmlLanguageFile lang) throws CmsException
4	CmsNewExplorerFileList.java	public byte[] getContent (CmsObject cms, String templateFile, String elementName, Hashtable parameters) throws CmsException
5	CmsTaskAction.java	public static void due (CmsObject cms, int taskid, String timeoutString) throws CmsException

6	CmsTaskContentDetail.java	public byte[] getContent (CmsObject cms, String templateFile, String elementName, Hashtable parameters, String templateSelector) throws CmsException
7	CmsTaskContentDetail.java	public byte[] getContent(CmsObject cms, String templateFile, String elementName, Hashtable parameters, String templateSelector) throws CmsException
8	CmsTaskList.java	public Object handleSpecialWorkplaceTag (CmsObject cms, Element n, A_CmsXmlContent doc, Object callingObject, Hashtable parameters, CmsXmlLanguageFile lang) throws CmsException
9	Utils	-public static String getNiceDate(CmsObject cms, long time) - public static String getNiceShortDate(CmsObject cms, long time)

Table 3. Summary of modifications for Date localization

In the above table, in all the classes from No.2 to No. 8, the change was made at the point where they call getNiceShortDate() of the Utils class. Before the modification, the call was made with only one argument i.e. getNiceShortDate(long time). Now they need to call it with two arguments i.e. getNiceShortDate(CmsObjec cms, long time) because of the change made in the class Utils.

The class Utils is the one modified a little bit more than others. The changes in summarized form were as follows:

- The public static String getNiceDate(long time) was changed to public static String getNiceDate(CmsObject cms, long time) and the public static String getNiceShortDate(long time) was changed to public static String getNiceShortDate(CmsObject cms, long time).
- The two classes written for the conversion of Gregorian to Ethiopic dates: com.opencms.boot.Ethiopic and com.opencms.boot.Day were imported into Utils.
- The code that was added in the getNiceDate and getNiceShortDate methods is displayed in Annexe E.

With these modifications now opencms knows to use Ethiopian date and time counting system.

Besides the above classes that are modified, there are other classes that store and access dates. These are not included in the modification because they do not display dates to the user.

Another way of accomplishing the same result was to create a CmsObject in the Utils class and to use it within getNiceDate and getNiceShortDate methods. Consequently this leads to no modification of any other classes. But creating a CmsObject needs to initialize it as follows:

```
CmsObject cms = new CmsObject();
Cms.init(broker, req, resp, user, currentGroup, currentProjectId,
streaming, elementCache, sessionStorage, directoryTranslator,
fileTranslator);
```

But when we look into init signature:

```
public void init(I_CmsResourceBroker broker, I_CmsRequest req,
I_CmsResponse resp, String user, String currentGroup, int
currentProjectId, boolean streaming, CmsElementCache elementCache,
CmsCoreSession sessionStorage, CmsResourceTranslator
directoryTranslator, CmsResourceTranslator fileTranslator) throws
CmsException
```

it needs a lot of objects in order to initialize CmsObject. This became a little bit complicated.

Building the source code with Apache-ant building tool

After the modification, OpenCms is now a new kind of OpenCms that exists nowhere else in the world. So it was needed to recompile it to get the binary form of OpenCms. The compilation was accomplished by an apache-ant version 1.6.1-building tool, which is available freely at the apache website. This tool is like the make command of the unix operating system, “but with out makes wrinkles”.

For the installation of ant and its usage see Annexe H.

5.5.2 Adding Ethiopian Calendar

The second task under the localization of OpenCms is to add Ethiopian Calendar.

The purpose of adding Ethiopian calendar is to make the calendar available over the Internet or intranet whenever an individual or company uses OpenCms as its content management system.

For the integration of Ethiopian calendar into OpenCms the two java classes (Ethiopic.class and day.class) developed for the localization of date and time, needed to be stored at the right place in directory structure of the operating system where the web server can find them.

This predefined place is found to be the directory CLASSES which is Under {TOMCAT_HOME}\Webapps\OpenCms\WebInf\classes. In this specific installation of the web server the place is c:\tomcat\ Webapps\OpenCms\WebInf\classes.

But storing the compiled java class in the right place in the file system of the underlined operating system is not by it self integration. It was also needed to write a java server page (JSP), which can accesses and use this classes. So, a jsp file called ethiopic.jsp was written and put in the root directory of the OpenCms Virtual File System (VFS). Now it is possible to link to this jsp file in order to use the Ethiopic calendar in OpenCms or to give the url of this file on an internet browser to use the calendar on the Internet or Intranet. Listing of Ethiopic.jsp exists in Annexe J.

5.5.3 **Localizing the built in calendar**

The third work under the customization of OpenCms is to localize the built in calendar of OpenCms. This calendar is displayed when a user tries to create a workflow in OpenCms. Since the built in calendar is a Gregorian calendar, it needs to be customized so that it displays Ethiopic calendar whenever the user prefers to work with Amharic language.

This time the localization does not require using the two-java classes created for the date and time localization above. But it required modifying template and java script files by which the built in calendar is written.

Since the built in calendar is written in java script, it is a must to have a Gregorian to Ethiopic converter in JavaScript. Consequently, a converter in JavaScript was developed. More over the template file that displays the calendar also modified. The complete listing of the changed files exists on Annexe F and Annexe G. The following table shows the summary of the modifications made.

No.	File	Inserted	Modified
1	Task_content_new	-	Delete onload=”fillDate()” from the body tag
2	Timer	-Var aMonthe()=new Array() and its assignements -pagumen is inserted -call to fillYear()	-onload=”sel()”
3	Opencms_timer.js	-Sel() -isLeapYear() -isEthiopicLeapYear() -GregorianToEthiopic -EthiopicToFixed() -FixedToEthiopic() -getEthiopicDayOfWeek() -fillYear() -yalenbetken -yalenbetwor -yalenbetamet	-selectDays() -checkTimerDate() -writeSel()
4	Workplace_am.properties (The resource file were the Amharic translation text exits.)	Calendar.months.pag = ጳጉሜን	

Table 4. Summary of modification for localization of built-in calendar

5.5.4 Localization of language – translating the OpenCms Workplace into Amharic

This section discusses about the activities that were required to translate the interface (workplace) of OpenCms into Amharic.

The following activities were performed by the AAU varsityNet team concerning the translation:

A. Create a new module:

   OpenCms works with modules. So the first step was to create a module for the Amharic translation by the name workplace_am.properties. “am” is the ISO code for Amharic.

B. Adjusting of the basic folder structure:

   During the creation of the module, OpenCms creates a directory structure(org.locales.am/classes/org/locales/am) in its VFS. Later on this structure should be changed to org.locales.am/classes/com/workplace/

C. Creating the locale file:

   This is the step needed to insert the Amharic translation into the resource file workplace_am.properties created in step one.

D. Testing the new locale

The last step is to test whether the above steps made OpenCms to work with Amharic language or not.

Out of the above activities needed to translate the interface of OpenCms, the following activities were the responsibilities of the author of this document

- Translating half of the words/phrases, which are integrated at the third step. During the translation, an English-Amharic dictionary and other English-English dictionaries were used. In addition Daniel Yacob's translation, which he made for linux, was a primary source.
- Preparing the operating system environment and adjusting the value of the encoding variable in some of the OpenCms files.

Under the second activity the first thing done was to set a system environment variable CATALINA_OPTS with a value of –Dfile.encoding = UTF-8. The next one is to make the value of the **encoding** variable equal to **UTF-8** in the following files:

- opencms.properties- The configuration file
- opencms.ori – Similar to opencms.properties

- registry.xml- The registry file
- web.xml – The application deployment file
- cmsshell.sh- A batch program for unix operating system
- cmsshell.bat – A batch program for windows operating system
- opencms.tld-
- web-app_2_3.dtd –
- Workplace_am.properties – the resource file for Amharic language

With out doing all these changes, trying to use Amharic language will display a "?".

Again, doing all these and trying to use Amharic language will display strange characters instead of Amharic characters. See the following picture.

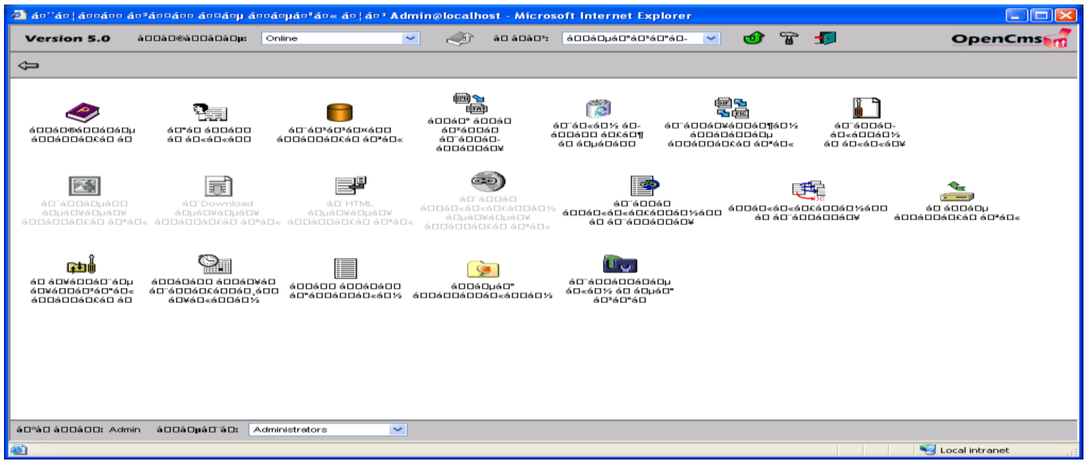


Figure 3. Before native2ascii command

This problem is not from OpenCms. It is the problem of the servlet engine Tomcat, which is based on java. Tomcat presents the characters in the Amharic resource file(workplace_am.properties) as ascii code. So there was a need to convert the workplace_am.properties file encoding i.e. UTF-8 to ASCII through java command. For this purpose java has a command called native2ascii, which was used as follows:

```
Native2ascii –encoding utf8 C:\jakarta-tomcat-4.1.27\webapps\opencms\WEB-INF\classes\com\opencms\workplace\workplace_am.properties C:\jakarta-tomcat-4.1.27\webapps\opencms\WEB-INF\classes\com\opencms\workplace\workplace_am.prop.
```

After the issuance of this command the converted file workplace_am.prop will be created in the same place as the original file workplace_am.properties. Since OpenCms needs the resource file by the name workplace_am.properties, workplace_am.prop was renamed back to workplace_am.properties.

One important thing worth to mention here is that whenever the workplace_am.properties is modified, it is a must to issue the native2ascii command as above. After issuing this command, to see the change, it is important always to stop and start the webserver.

5.6 Verification and validation

The selected software, opencms, was tested for the functionalities specified on the requirement document. (See Annexe B)

For the development of the calendar program the following verification and validation steps were taken:

Each method exist in the Ethiopic class now, was tested as a standalone program. After these unit testes over, an integration test was performed. For this, a test class was generated (See Annexe J). As a result the Gregorian date to Ethiopic date conversion program was proved to work as a stand-alone program. For the integration test 1994, 1995 and 1996 calendars data were used.

Later on when the integration test was finished, the conversion program was integrated into the OpenCms system. After the integration, a verification that checks whether the integration is successful or not was made by recompiling OpenCms from the modified sourcecode. Next was to make all necessary modification in order to localize OpenCms for the date and time counting system of Ethiopia. After each modification, OpenCms was built from the modified source code. And then it was setup from the new binary form. It was found that OpenCms is able to work with Ethiopian dates and time.

For the translation part, before the translations are integrated with the system they were checked and corrected by Dr. Solomon Atnafu. After they are inserted in to the system a walk-through test was used to see whether the translation is meaningful when the system is looked in action.

For the localization of the built in calendar, at each modification point it was needed to setup OpenCms again and check for the modification result.

5.7 Critical evaluation

This product fully satisfies the requirements of the project. Now OpenCms can work with Amharic language and date and time counting system of Ethiopia i.e. it is localized for Ethiopian language, culture and conventions.

In addition, the success of the project showed that open source software beneficiaries in two ways: first, there is no need to develop a program from scratch if it were developed some where else in the world and it is an open source software. Second, without having a big investment, we can have the latest technology with our own language and culture.

This product did not show other localization issues like localizing sounds, pictures, currency symbol, number format, etc because this things are not included in the original product it self.

Still the user interface of the workplace is mixed with English. This is because OpenCms does not want to change the name of some of the files. They are like system property. For the new one also it was not possible to store their name in Amharic.

There is a problem of displaying Amharic characters on a drop down list box using Internet Explorer version 6.0. When displayed on Mozila version 1.4, Amharic texts in the drop down list box are correctly displayed.

This product will have a positive impact if implemented in any business and public organizations. This is because of Internet. Internet is becoming an increasingly vital tool in the information society. More people are going online to conduct such day-to-day activities as business transactions, personal correspondence, research and information gathering, and shopping. Each year, being digitally connected becomes ever more critical to economic, educational, and social advancement. Now that a large number of people around the world regularly use the Internet to conduct daily activities, organizations that lack those localized content management products will be at a growing disadvantage.

5.8 Lesson Learned

In this project the most important lesson learned is that open source software are the way to cope with the current information technology. If we can get software with a minimum total cost of ownership and a software which can easily be adaptable to our culture and conventions, the computers, which are seeing as a sophisticated typewriter now, will be more useful and productive in this country.

Beside this general lesson, the following specific lessons are learned at the different steps of the development of the system

- ❖ During Installation

When installations are performed, sometimes the document, which holds the steps for the installation are not enough. As a result stacking may be frequent. Such a things are happened when the installation documentation assumes some background knowledge of the person. For example during the installation of tomcat, one of the steps needed was setting of an environment variable. How to set the environment variable was not included in the documentation. This is because, setting an environment variable is a knowledge found in the documentation of a given operating system.

❖ During translation

Translation needs two things: One is a good amount of vocabulary of the target language. Second, it needs a good understanding of the system. Knowing only the language that the original content is written in is not enough for localization. The key issue is not the words and phrases but the meaning. In technical translation the literal translation of words and phrases is not important. What is important is the transfer of information to a specific target group that has no access to the original content.

❖ During customization

During customization, the main thing learned is that it is good to get the map of the software. Otherwise getting into a software that are big like OpenCms is like getting into a big city to find a person without having an address of that person.

❖ Mailing list

One of the big sources for open source software is a mailing list around that open source software. Most of the time it is very crucial to understand the standard of communication in a given mailing list. Sometimes what we write thinking that it is right may be an offense for the members of the mailing list. Sometimes no body answers for a question.

- During modification

After modification of java classes we need to build opencms from the source code using Ant and then setup OpenCms again. So whenever there are changes, changing in batch mode is very important. Otherwise every building and setup in average will take more than 8 minutes.

After modification of java scripts and template files we do not need to build opencms from the source code. But we need to setup it again. Other wise it does not see the changes made in this kind of files. In average this takes 5 minutes.

- Documentation

When compared to proprietary software, many say that open source software are half documented. Again many say that the best document for an open source software is the source code it self. So for startup, reading that half document is important. But for finishing, the source code is the most important document.

6 Conclusions

Information is power. This implies that we should properly manage the information that we have. Unfortunately, most documents in Ethiopian government offices are not properly managed. This is because of the traditional method of document management and exchange. The traditional method of document management is very time consuming and tiresome and it is susceptible for easily damage and lost of documents.

CMS can improve the management of documents and its content by providing mechanisms to exchange and manage information and documents, send and receive e-mail in organized form, conduct discussion forums and chatting comfortably, archive documents, upload and download files, search documents with their names and contents, schedule tasks, provide report template, maintain history of documents, manage workflow, browse documents and maintain websites. However, unless it is localized to the working conventions of Ethiopian the use of such software will be limited.

In this project we took OpenCms a powerful open source content management system and localized it. Using open source enabled us to reuse the work of hundred of people around the world with minimum cost.

The localized OpenCms can now be used for Ethiopian government offices as well as other offices after some minor improvements.

The system could be extended in a further project to provide translation for other local government languages of Ethiopia.

Regarding date and time localization as seen in this project, it needs to modify the core code of OpenCms. Continuing modifying OpenCms for all the cultures that have their own calendar system is not a visible way. Instead adding a module so that OpenCMs will have a single point where all localization things can be done without entering into modifying the code is one of way improving this work.

The addition of a translator module into OpenCms so that content developed in one region with the language of that region can be used in another region with a different language deserves future research project.

Developing searching engine for Ethiopic texts, storing resource names like file, folder, web page, link names in Ethiopic characters, and developing a calendar that can tracks events are other subjects that could provide huge improvements for the product developed in this project.

7 References

[1] Alan Wood

Test for Unicode support in Web browsers
Ethiopic
http://www.alanwood.net/unicode/ethiopic.html

[2] Alan Wood

Creating Multilingual Web Pages:
Unicode Support in HTML, HTML Editors and Web Browsers
http://www.alanwood.net/unicode/

[3] Alan Wood

Setting up Windows Internet Explorer 5, 5.5 and 6
for Multilingual and Unicode Support
http://www.alanwood.net/unicode/explorer.html

[4] Dawit Bekele

The Development and dissemination of Ethiopic Standards

and Software Localization for Ethiopa

[5] Dswigger

Localizing software application

http://www.codeproject.com/asp/localising.asp

[6] Document Management

The Quill Consultancy>>NewLetter No. 1-2004
Http://www.quill.com.ay/news/letter
/Document_mangement.htm

[7] The Ethiopic Unicode Resource Page

http://www.abyssiniacybergateway.net/fidel/unicode/

[8] Gf Zemene font

ftp://ftp.ethiopic.org/pub/fonts/TrueType/gfzemenu.ttf

[9] Internationalization

www.w3.org/internationale/tutorials/tutorial_char_enc.html#choos

ing

[10] Jeff Friesen

Internationalize your software

http://www.javaworld.com/javaworld/jw-01-1999/jw-01-internationalize.html

[11] Michael Suodenjoki
Introduction to internationalization and localization
Globalization of software applications

[12] OpenCms Documentation Version 5.0
http://www.opencms.org

[13] OpenCms Mailing list Archive
http://www.opencms.org/opencms/en/development/mailinglist-archive.html

[14] OpenCms Mailing list
http://www.opencms.org/opencms/en/development/mailinglist.html

[15] Per N. Dohler
Facets of Software localization
A translators view
http://www.accurapid.com/journal/softloc.htm

[16] Spol. Sr.o.
Why is software localization so demanding
http://localization.exe.sk/english/demanding.asp

[17] Tex Texin
Compelling example of Unicode usage for business applications

[18] http://www.webOpedia.com

Annexes

Annexe A. Requirement Analysis Document

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.[1]

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.

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.

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

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.

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.

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.

Definitions

Content search
The ability of a system to search through text to match a group of characters.
Optical Character Recognition (OCR)
Optical character recognition refers to the branch of computer science that involves reading text from paper and translating the images into a form that the computer can manipulate (for example, into ASCII codes).
Open source software/Free Software
Software which allow users' to access the source code, freely run, copy, distribute, study, change and improve the software
Total cost of ownership (TCO)

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

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. [2]

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.

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

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

do so. Therefore, almost all organizations have record offices that are so packed that it is virtually impossible to search and find documents.

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. [3]

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 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. [4]

The public offices do not have the necessary know-how or the technology to process the available information analytically. Comparisons 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]

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

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

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. [5]

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.

Proposed system

Overview

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

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.

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.

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.

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,		This use case describes the

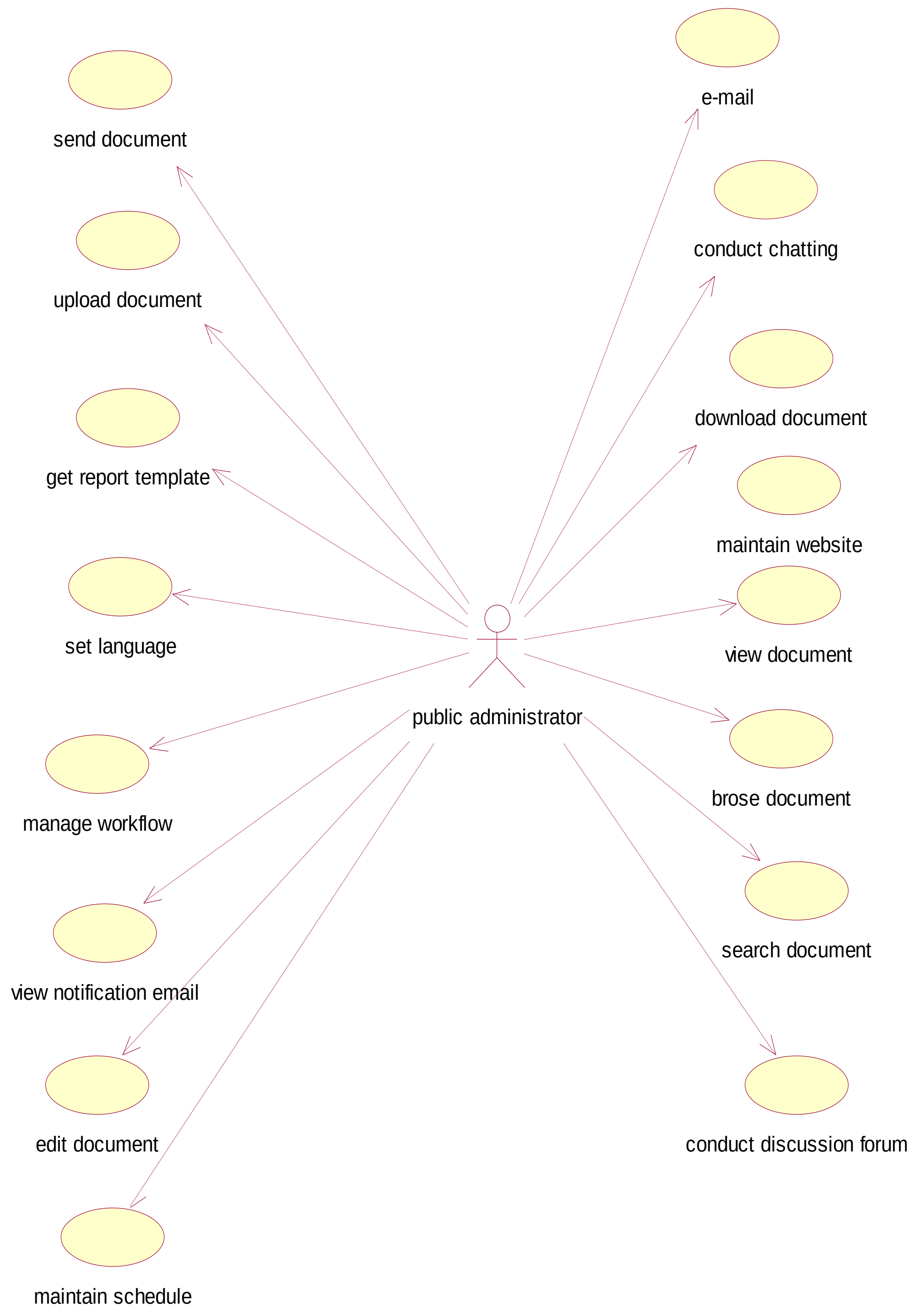
public administrator, Server administrator	Search document	activities done by the system when the user wants to locate a document or files, but does not remember the actual location or name of the file. The search can be done by file name, by content, by size and by creation or modified date.
Public 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 administrator	Set language	This use case describes the activities done by the system when a user wants 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

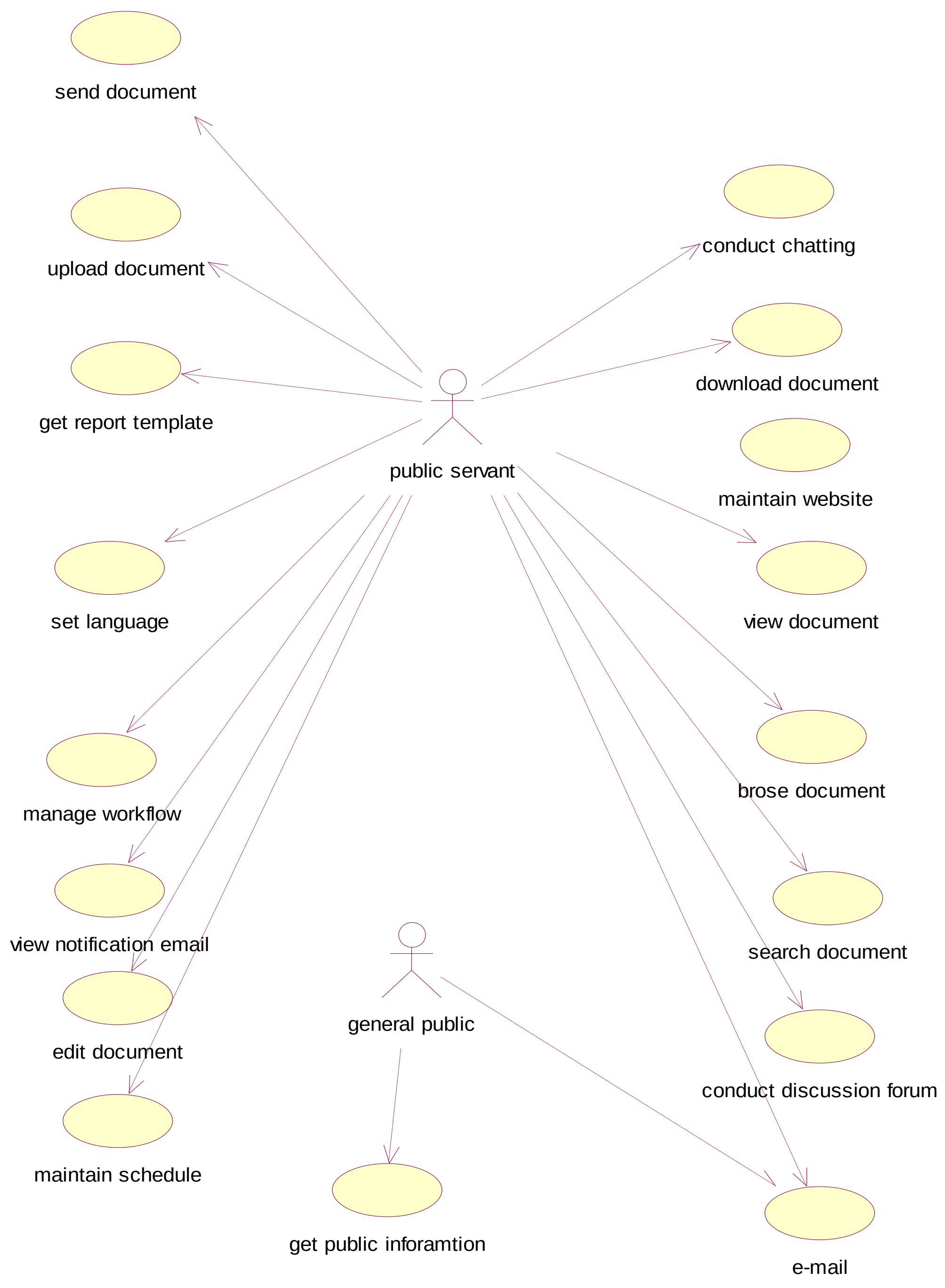
		(What you see is what you get) editor tools.
Public servant, public administrator, server administrator	View document history	This use case describes the activities done by the system when a user wants to know when and who edited, formatted, viewed, deleted, archived and recovered a particular file or 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
general public
get public inforamtion
conduct chatting
download document
maintain website
view document
brose document
search document
conduct discussion forum
e-mail

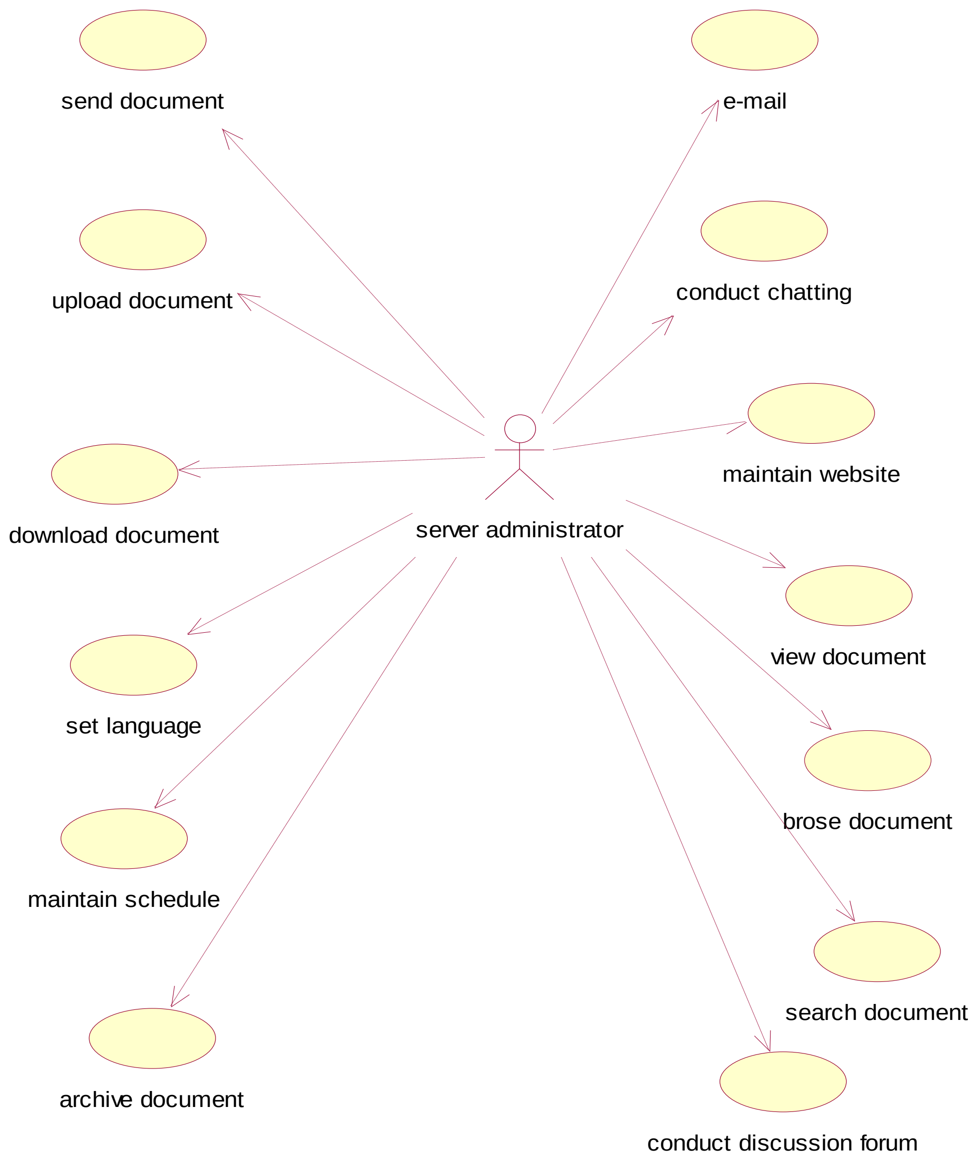


Fig. 1. Use case diagram

Annexe B. Review of Content Management system

Review of Open Source Content and Document Management Software Products

1. Introduction

It is obvious that Organization's, institution's, or companies' information resides in documents. Documents may contain any kind of information starting from company's rules and regulations up to a very high secrete of the company. If we consider a kebele administration, every minute created for each meeting, every reports prepared, ownership information, contracts, etc are stored in a document. Therefore these documents are very important for the kebele administration as well as the people leaves in the kebele. So, the kebele is required to use and store the documents with great care. But due to reasons like shortage of spaces, fire, rain, rodents, etc it is not a simple matter to manage documents manually. Especially, when they become large in kind and number.

The need for greater efficiency in handling documents fueled the rapid development of document management software systems. There is much software that is developed in the last decades that can be used to manage documents as well as their contents. These software are both open source and proprietary.

The main objective of this study is to identify those open source content management system software that are most adequate for the development of multilingual content management system or that are customizable enough to adapt for the Ethiopian context.

2. Overview of content and document management software

This section defines important terminologies needed through out the document. Based on the definitions it will show the similarities and differences between Document Management Systems versus Content Management Systems.

2.1 Definition

2.1.1 Document and Content of a document

Document: A document is any container of coherent information, which has been assembled for human understanding.
Content: Any thing stored in a document

2.1.2 Document management system (DMS) and Content management system (CMS)

Document Management System:
Document management is the automated control of electronic documents- page images, spreadsheets, word processing documents, and complex, compound documents- through their entire life cycle within an organization, from initial creation to final archiving.

Document management allows organizations to exert greater control over the production, storage, and distribution of documents, yielding greater efficiencies in the ability to reuse information, to control a document through a workflow process, and to reduce product cycle times. The full range of function that a document management system may perform includes document identification, storage and retrieval, tracking, version control, workflow management, and presentation. [Gary Cleveland]

Content Management System:
A Content Management Systems, like a document management systems, they are essentially a collection of business rules and editorial processes. However, the rules and the process are around content instead of the entire document.

2.1.3 Relationship between Content management Software and document management software

Since a document (such as a technical manual) may contain one or more units of digital information - or content - a document is in essence comprised of content. Since both a CMS and a DMS enable information to be managed according to rules, processes and workflows, the main differentiation between the two products becomes the granularity of management of the digital information a CMS offers when compared to a DMS.

A DMS is concerned with a Document in its entirety and less interested in what the document contains. A CMS effectively manages at a micro level the individual pieces/units of info that go to making up a document or web page.
All in all, Document Management system is an important precursor to Content Management system.

2.2 General features of content management software

This section explains the general functionalities that different CMS's provide. Many open source Content Management System provide the following functionalities:
Document repository:
This is the main purpose of any content management system. Most CMS's use databases as their repository. Others use flat files or XML files.

Multilanguage support:
This is the functionality from cms's to make a site to work with many languages. To support multiple languages a content needs to be an internationalized- able to store its contents in multibyte site characterset- software

Searching:
Functionality used to find documents by their content, and index fields such as date, file name, file type, and size.

Workflow:

Workflow is an IT technology, which uses electronic systems to manage and monitor business processes. It allows the flow of work between individuals and/or departments to be defined and tracked. Although Documents are often used as a medium for transporting information in a Workflow system, it is mostly associated with Document Management where the Workflow system is used to track the process of creating and reviewing and distributing Documents.

Security:

Security is the feature used to secure the system. It is about, user permission and rights, supporting LDAP, supporting SSL, plug in authentication, login history, session management and so on.

Versioning:

This helps to store a previous version of a changed document. In a system where there is such functionality it is possible to return back to an older version of content.

Localization: The feature used to adapt the software to ones local condition. It includes settings for formatting of currency, date, time and other local specific values.

Browser-enabled work environment: The feature used to create, update, and manage any page element through the generic web browser.

Template mechanism: Used to present page elements to the user of the web page. It is possible to create one template for each content type, or more if it is needed to present contents in various ways.

Web site creation and maintenance:

In content management system there are many features used to create and maintain a web site. Some of the common features include are poll, survey, authoring tools, discussion forum, link management, creating document for multiple targets, graphics and multimedia management, personalization, and templating

Plugin API:

This is the feature used to expand the system in order to incorporate a new functionality in to the software.

3 Challenge of introducing Content management system in an organization

3.1 Resistance from organization employees

Document Management is a core business activity and can therefore impact large parts of an organization. The introduction of a content management system to an organization immediately exposes where the weakness is in managing the documents. In order to minimize the disruption, content management should be introduced in phases and with careful change management procedures.

3.2 Selecting the right software

Since introducing content management software is a major decision and it is difficult to select the right software for all kinds of users. Some of the reasons that will make selecting a CMS software challenging are making the following mistakes:

- Letting a software vendor telling what a customer need
- Not establishing a broad-based selection team

- Not understanding the total cost of ownership
- Not thinking strategically about content management
- Not understanding the parameters of web content management.

Besides these, to select the right content management system software it needs to decide on the kind of the site, the features and functionality needed, the look and feel of the site, the content of the site and on going maintenance and renewal of the site

4 Some of the Open source content management systems software

As indicated above, there are hundreds of open source content management software. Out of these software that exist in the market today, eight of them are selected for this study. This selection is based on the following criteria:

- Whether the software is supporting multiple languages or not.
- The support type they have: do they have commercial support, mailing list, user friendly community, etc
- Their system requirement: This is a criteria that looks in to whether the software require a commercial software or not? If it requires, it wouldn't be selected
- The features they have: whether the software is rich with its features or not?
- Content management framework: Whether they have the framework to develop a new CMS based on them or not?

Using the above criteria the software selected are: Typo3, EZ publish, Bricolage, OpenCMS, Midgard, Webgui, Zope, and Twiki.

4.1 TYPO3

TYPO3 is open-source content-management and publishing system. Main features include are authoring tools, WYSIWYG editor, image management, File and media management, scheduling system, Importing of other files like word documents, version control, Multilanguage support, personalization, plug ins, search engine, user management and permission control, page view statistics, workflow engine, task center for workgroup collaboration, template management, management of multiple, independent websites in the same installation, and support ssl.

Typo3 is written in Perl and uses. Apache/mod_perl web server. Typo3 uses any relational database to store its content.

4.2 EZ Publish

EZ Publish is an open-source content-management and development framework. As a content management system, it has the following features template engine, role based permission, work flows and extensions, translation and localization, support cross platform, version control, Multilanguage support, integrated search engine and ecommerce functionality, features that helps collaboration, and plug in editor for simplified content editing.

EZ publish comes with a standard setup and the most common features of a web site are ready to use: Administrative interface, article editing and publishing, discussion forums, e-commerce, and collaboration and instant messaging system.

EZ Publish is written in PHP, and uses Apache web server. EZ Publish uses the PostgreSQL/MySQL relational database to store its content.

4.3 Bricolage

Bricolage is an open-source content-management and publishing system. Features include configurable administration, workflow, permissions, templating, distribution, and document management.

Bricolage is written in Perl using HTML::Mason and Apache/mod_perl. Bricolage uses the PostgreSQL relational database to store content

4.4 OpenCMS

OpenCMS is a Java/xml based open source content management system. It includes the following major features: Browser-enabled work environment, images and other binary download files management (Asset management), integrated user management and permission system, workflow and task management, WYSIWYG editor, Internationalization support, version control, template mechanism, Multi-language support, Online-help system, dynamic and static content publishing, personalization, caching mechanism, module mechanism for extension, SSL support, scheduling system, import and export of content, distributed object architecture, clustering for load balancing and failover and meta information attachment with various resources .

OpenCMS is written in Java. It uses Apache Tomcat web server. OpenCMS uses Varieties of relational database management system to store its contents

4.5 Midgard

It is a content management framework by which other content management application can be developed. Features include are: Internationalization, Templating sysem, graph structure for pages, articles, formatting engine, managing users, personalization, calendaring, and attachments.

Midgard is written in PHP. It uses Apache webserver. Midgard stores its content using only MySQL database.

4.6 WebGUI

WebGui, a content management system and content management framework, has a pluggable object architecture, which is used for extensibility. Many built in objects include are: Message boards, SQL report, polls, Download manager, user contribution system as a web log and photo gallery, FAQs, Link directory, syndicated news, Events calendar, articles, pluggable authentication, integrated search engine, internationalization and localization support, template management, image management, and trash system as a failsafe for system administrator.

WebGUI is written in perl. It uses apache or IIS web servers. WebGui can use different relational database management system to store its contents

4.7 Zope

Zope is a content management system as well as a framework for building web application. Zope consists of several different components that work together to help building web applications. Zope comes with:

A Web server, a web based interface, an object database, a mechanism to work with other relational databases, and Scripting languages that allows writing web application.

Some of its features are: Asset management, version control, templating, document management, versioning, and workflow management.

Zope is written in python and c. It uses its own webserver called zserver or other web servers like IIS or apache. Zope has its own database called ZODB. It can also be used with different relational database management system like MySQL, PostgresSQL, and Oracle.

4.8 Twiki

Twiki is a web based collaboration platform. It can be used to run a project development space, a document management system, a knowledge base, or any other groupware tool, on an intranet or on the Internet. Web content can be created collaboratively by using just a browser. Developers can create new web applications based on *a* Plugin API.

Twiki supports the following features: Full text search with or without regular expression, E-mail notification, Upload and download any file as an attachment, version control, template and skin management, access control based on groups and users, and syndication.

Twiki is written in Perl using HTML::Mason and Apache/mod_perl webserver. Twiki uses the PostgreSQL relational database to store content.

5 Comparison of Open source CMS software

To compare open source content management systems it is possible to use their system requirement, security feature, support, general features, and built-in applications they provide.

Product	**WebGui**	**OpenCMS**	**Bricolage**	**Ez publish**	**Zope**	**Typo3**
System Requirement						
Language	Perl	Java	Perl	PHP	Phyton, c	Perl
Operating System	Any	Any	Linux		Any	Any
Web Server	Any	Tomcat	Apache/ Mod_perl	Apache	Zserver, IIS, Apache	Apache mode_pe
Database	MySQL, Postgres	Any	PostgreSQL	MySQL/ Postgresql	ZODB	Any
License	GPL	GPL	GPL	GPL	GPL	GPL
Security						
Granular Privileges	Yes	Yes	Yes	Yes	Yes	yes
LDAP support	Yes	-	-	-	Yes	-
Pluggable Authentication	Yes	-	-	yes	-	-
Versioning	No	Yes	Yes	Yes	Yes	Yes
Content Approval	No	Yes	No	Yes	Yes	Yes
Audit Trail	Yes	-	-	-	-	-
Login History	Yes	-	-	-	-	-
Session Management	Yes	Yes	Yes	Yes	Yes	Yes
Support						
Commercial Support	Yes	Yes	Yes	Yes	Yes	Yes
Developer Community	Yes	Yes	Yes	Yes	Yes	Yes
Online Help	Yes	Yes	Yes	Yes	Yes	Yes
Third-party Developers	Yes	-	-	-	yes	-
Certification Program	Yes	-	-	-	-	-
Features						
Template Management	Yes	Yes	Yes	Yes	Yes	Yes
Workflow engine	-	Yes	Yes	Yes	Yes	Yes
Image management	Yes	Yes	Yes	Yes	Yes	Yes
Internationalization	Yes	Yes	Yes	Yes	Yes	Yes
WYSIWYG Editor	Yes	Yes	-	No	No	yes
Clipboard	Yes	Yes	Yes	Yes	Yes	Yes
Scheduling	Yes	Yes	-	-	-	yes
Built-in Applications						

Discussion/Forum	Yes	Yes	Yes	Yes	Yes	Yes
Web Log	Yes	-	-	-	yes	-
User Contribution	Yes	Yes	Yes	Yes	Yes	Yes
FAQ management	Yes	-	-	-	-	-
File Distribution	Yes	-	-	-	-	-
Events Calendar	Yes	Yes	Yes	Yes	Yes	Yes
Polls	Yes	Yes	-	Yes	Yes	Yes
Surveys	Yes	Yes	-	Yes	Yes	Yes
E-Mail notification	Yes	-	-	-	-	Yes
Guest Book	Yes	-	-	-	-	-
Group Ware	No	-	-	-	Yes	-
Syndicated Content (RSS)	Yes	No	No	No	Yes	No
Tests/ Quizzes	Yes	-	-	-	-	-
Personalization		Yes				Yes
Document Management	No	No	Yes	No	Yes	No
Chat	No	-	-	-	Yes	-

6 Criteria for selecting globally ready (multilingual) content management system

A truly “global” content management solution must include multilingual interfaces for all critical applications, *localization-specific workflows out-of-the-box*, integrated language/locale menus for all supported authoring tools, and an underlying metadata structure that accounts for language and regional differences.

When evaluating global-readiness of a content management system, there are five key considerations:

1. Are the Content Management System and all supporting databases fully internationalized?
2. Does the system allow you to identify and tag content by language and locale?
3. Does the system link source language content and translated content?
4. If translation vendor are used for translation purpose, are they familiar with the content management environment?
5. Is the workflow capable of incorporating translation steps? Content management systems are evaluated on many different parameters, but a robust and flexible workflow engine is a “must have” requirement.

These key questions will help effectively evaluate the “global-readiness” of a content management solution. Equally important to the assessment of these capabilities however, is deciding how and when to leverage language management tools and globalization technology. An effective content management solution will take advantage of state-of-the-art language management technology – which includes translation memory, terminology management and other tools that improve the efficiency and accuracy of human translation.

7 Conclusion

In Ethiopia, most of the time and in most organization, documents are created by using pen and paper or manual typewriter and paper. More importantly, documents are not managed electronically. This is simply the implication that documents are highly unused in this country. The difficult process of creating, storing and retrieving documents expresses this.

The problem of under utilization of documents will likely grow in the future. Instead of decreasing, this trend is increasing because the amount of documents produced will grow and more people with low educational level will come to use the computer through the woredanet project, and finally creating and using of electronic documents without having a right content management system will increase.

The solution to this problem is to have a multilingual content management system, which helps even less educated people to create, store and organize their documents through their local languages. For this purpose, one of the above-mentioned software can be customized and used in the Ethiopian context.

8 Glossary

API

Abbreviation of *application program interface*, a set of routines, protocols, and tools for building software applications. A good API makes it easier to develop a program by providing all the building blocks. A programmer puts the blocks together.

Download

To copy data (usually an entire file) from a main source to a peripheral device. The term is often used to describe the process of copying a file from an online service or *bulletin board service* (BBS) to one's own computer. Downloading can also refer to copying a file from a network file server to a computer on the network.

GPL

Short for *General **P**ublic **L**icense,* the license that accompanies some open source software that details how the software and its accompany source code can be freely copied, distributed and modified. The most widespread use of *GPL* is in reference to the GNU GPL, which is commonly abbreviated simply as *GPL* when it is understood that the term refers to the GNU GPL. One of the basic tenets of the GPL is that anyone who acquires the material must make it available to anyone else under the same licensing agreement.

LAMP technology

Short for ***L**inux, **A**pache, **M**ySQL and **P**HP*, an open-source Web development platform that uses Linux as the operating system Apache as the Web server, MySQL as the RDBMS and PHP as the object-oriented scripting language. Perl or Python is often substituted for PHP. LAMP has become a de facto development standard.

Open source

Generically, *open source* refers to a program in which the source code is available to the general public for use and/or modification from its original design free of charge, i.e. open.

Plugins

A hardware or software module that adds a specific feature or service to a larger system

RSS

Short for ***R**DF(***R**esourced Description Framework) ***S**ite **S**ummary or **R**ich **S**ite **S**ummary,* an XML format for syndicating Web content. A Web site that wants to allow other sites to publish some of its content creates an RSS document and registers the document with an RSS publisher

Syndication

The sharing the content among different Web sites

Upload

To transmit data from a computer to a bulletin board service, mainframe, or network

Blogs

Short for *Web log,* a blog is a Web page that serves as a publicly-accessible personal journal for an individual. Typically updated daily, blogs often reflect the personality of the author.

Wiki

A collaborative Web site comprised of the perpetual collective work of many authors. Similar to a blog in structure and logic, a wiki allows anyone to edit, delete or modify content that has been placed on the Web site using a browse interface, including the work of previous authors.

WYSIWYG

Short for *what you see is what you get*

SGML

Short for ***S**tandard **G**eneralized **M**arkup **L**anguage,* a system for organization and tagging elements of a document.

XML

Short for ***E**xtensible **M**arkup **L**anguage,* a specification developed by the W3C, XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definiti0on, transmission, validation, and interpretation of data between applications and between organizations.

SOAP

Short for ***S**imple **O**bject **A**ccess **P**rotocol,* a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and may be ransported using a variety of Internet protocols, including SMTP,MIME, and HTTP.

9 Reference

http://www.documentmanagment.org.uk/pages/reference.htm.

http://www.document-manager.com/

http://www.internetwk.com/news/news1006-6.htm

http://nfocentrale.net/dmware/

http://www.cmswatch.com/Features/OpinionWatch/FeaturedOpinion/?feature_id=84

www.zope.org/

www.zope.com/

cmf.zope.org

www.zopelabs.com/

http://www.xope.org/Resources?ZopeIntro/

http://www.ez.no/products/ez_publish_3/key_features

http://www.ez.no

freshmea.net/projects/ezpublish/

http://www.sitepoint.com/article/917

www.typo3.com/
www.typo3org/
demo.typo3.com/
sourceforge.net/projects/typo3/
www.meridium.com/
http://www.cmsinfo.org.
http://www.cmsinfo.org/index.php3?sectoin_id=25
www.etranslate.com.au/3_2.asp
www.meridium.com
www.opencms.org/
freshmeat.net/projects/opencems/
http://www.cmsinfo.org/article.php3?story_id=421
http://www.bricolage.cc/
http://www.eweek.com/article2/0,3959,652977,00.asp
twiki.org/
www.infotoday.com/searcher/apr03/mattison.shtml

Annexe C. Plone/Zope installation

Note: This installation documentation is taken from the ploneBook-PDF.pdf

Requirements

Plone will install on any of the platforms that Zope supports: Windows, Mac OSX, Linux, most Unixes and Solaris. Windows 2000 requires writing to the registry in order to install the software, this may require more rights than you have.

Server

A higher performance computer will obviously make Plone perform better. Plone is not a toy. It requires horsepower and memory. In general, you shouldn't go into production with a machine slower than 1.5Ghz and less than 1GB of RAM if you are serving a large website. It works fine with setups as low as 500MHz and 64MB of memory for more modest sites, however.
For a base installation of Plone about 50Mb of hard drive space is required. If you already have installations of Zope or Python, then this can be dramatically reduced. You must also account for the Plone object database
which can grow to almost any size depending upon the amount of data you store.

Client

Plone only requires a web browser that can access the server. If users want to login, cookies must be enabled. JavaScript is not required but will provide a richer user experience. Required browsers for Plone 1.0:

- Internet Explorer 5.5 and up (6.0 and up recommended) ·
- Netscape 7.0 and up ·
- Mozilla 1.0 and up (1.4 and up recommended) ·
- Opera 7.0 and up (7.20 and up recommended) ·
- Konqueror 3.0 and up (There are some inconsistencies, but you will have to live with those until they are updated against the KHTML code base changes made in Safari)

·

- Safari 1.1 and up ·

Plone also is fully functional in the following browsers, but might look different from the original Plone look:

- Netscape 4.7 and up ·
- Internet Explorer 5.0 ·
- Internet Explorer 4.0 (not extensively tested, but should work well) ·
- Konqueror 2.x ·
- Lynx (text based) ·
- w3m (text based) ·
- AWeb ·
- links (text based, with optional graphics) ·
- Any browser that handles a basic set of HTML and form input + cookies (including most mobile/PDA browsers)

·

Downloading Plone

The latest Plone is always available at http://www.plone.org/download.

Installing using the Windows Installer

The Installer

The Windows installer automates the installation of Plone on Windows. Windows versions 9x, ME, NT 3.51+, 2000 and XP have been tested, but it may work on others. It is recommended you have administrator access on the computer you wish to install on. If you already have Zope or Python installed, you may want to look at installing the source seperately to save hard drive space. The installation includes extra packages and options, a pre-loaded database and other goodies.
The Plone installer for Windows can be downloaded from the Plone.org website, in the downloads section.
Once you have downloaded the installer, double click on the installer to begin the install.

The installer goes through the usual steps for installing software, follow the options at the bottom for "Next" or "Cancel". There is no need to discuss all the steps, most are self explanatory. When you get to the "Enter a
password" screen (shown below), you must enter a password. This will register a password for the "admin" user. You will need this password later, so make note of it. If you do lose this password you can enter a new one through the Plone Controller.
The installation takes somewhere around 3 minutes, depending upon the speed of your computer. A few tasks are performed at the end of the installation, such as compiling all the Python files. When the installation has finished, Plone is not started by default. If you leave the "Launch Plone Controller" box checked the Plone
Controller will be launched which will enable you to start Plone.

Annexe D. Ethiopic.java and Day.java

The first thing done during addition of calendar module and localization of date and time is to convert a c code written by Daniel yacob to java code. In the c code of Daniel Yacob there were 8 files. The files are written using the K&R styles of the 1970's.

The files and their purposes are explained in the following table:

No.	File	Purpose
1	Ethiopic.c	Contains functions to check whether a given date is a correct Ethiopian date, to covert greorian date to ethiopic date, to get last day of ethiopic months, to get Ethiopic day of a week, to get Ethiopian holiday and other accessory functions.
2	Gregorian.c	Contains functions that are used to convert Ethiopic dates to Gregorian dates, get day of a week, returns the number of days in a given Gregorian month, check the correctness of a Gregorian date and some other additional necessary functions.
3	edate.c	Used for handling dates
4	etime.c	Used for handling time
5	etime.data	Contains the following for both Amharic and Tgrigna languages -UTF8 day, month, holiday, year names -SERA day, month, holiday, year names -Transcribed day, month, holiday, year namees
6	etime.data.sera	- The same as etime.data but with out Amharic characters
7	etime.h	Defines all the functions prototype which are in the Gregorian.c and Ethiopic.c files. And also it defines a lot of constants
8	etimeex.h	Define character pointer arrays for Amharic and tigrigna sera month, day, year names, ISO639_3Names[];ISO639_2Names

Out of these files two java classes called Ethiopic.java and Day.java that contains all the necessary methods and data for the conversion of Gregorian dates to Ethiopic dates has been generated (See Annexe D).

Since java is somewhat similar to c++, converting c to java seems strait forward. But, in practical, there are many c statements that do not have a direct match in java. More importantly, since java does not support pointers, it was hard to convert them in to java supported data structures. Mechanisms used during the conversion are as follows:

- Preprocessor directives (#ifdef, #ifndef, #if, #include) were dropped
- Pointers
    i. Char * -> String []
    ii. int * - > int[]
    iii. Pointers passed to functions were converted to a class objects.
- Structures were directly converted to java classes
- Functions
    i. Maloc and free functions are discarded
    ii. All statements and functions concerning Tigrigna were dropped.
    iii. All functions declared in header files were transformed into public static functions
- Data
    i. In the Daniels Yacob program data of month, year, holiday, and day names were stored in a separate files in three formats. These are SERA names, Transcribed names and UTF8 names. In the converted java files these data are stored together with the method that manipulates them. How the UTF characters stored in the java files with out having a Unicode editor? See below under the title Encoding.
    ii. Data for Tigrigna were dropped

All data declared in header files are transformed to a public static constants in Ethiopic.java class

In the converted java program names of Amharic months, and days were necessary to store using the Amharic characters. But the available java editors couldn't work with a Unicode characters. As a result it was not possible to store the characters directly. So the technique of storing Unicode characters using their Unicode is used. Eg. "ሰኞ" was stored as "&#x1230;&#x12.."

```
/* This code is converted from What Daniel Yacob was written for Ethiopian
  Calendar in c.

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Library General Public
  License as published by the Free Software Foundation; either
  version 2 of the License, or (at your option) any later version.
```

```
   This library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   Library General Public License for more details.

   You should have received a copy of the GNU Library General Public
   License along with this library; if not, write to the Free
   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
*/
package calendarEth;
import java.text.*;
import java.util.*;
public class Ethiopic
{
        public static final int E_EPOCH = 2796;
        public static String[] ethDates = {"Ehud", "Segno", "Maksegno", "Erebu", "Hamus", "Arib",
 "Kidame"};
public static String[] ethUTFDates = {"???", "??", "????", "???", "???", "???", "???"};
        public static String[] ethMonths = {"Meskerem", "Tikmit", "Hidar", "Tahisas", "Tir",
"Yekatit", "Megabit","Miazia","Ginbot","Sene","Hamle","Nehase","Pagumen"};
        public static String[] ethUTFMonths = {"&#x1218;&#x1235;&#x12a8;&#x1228;&#x121d",
"&#x1325;&#x1245;&#x121d;&#x1275", "&#x1205;&#x12f3;&#x122d", "&#x1273;&#x1205;&#x1233;
&#x1235", "&#x1325;&#x122d", "&#x12e8;&#x12ab;&#x1272;&#x1275",

"&#x1218;&#x130b;&#x1262;&#x1275","&#x121a;&#x12eb;&#x12da;&#x12eb","&#x130d;&#x1295;
&#x1266;&#x1275","&#x1230;&#x1294","&#x1200;&#x121d;&#x120c","&#x1290;&#x1213;&#x1234",
"&#x1333;&#x1309;&#x121c;"};

        public static String[] ethYearNames = {"Matheos", "Markos", "Lukas", "Yuhanes"};
        public static String EthHolidays[] =
  {
    "New Years",               /* Meskerem  1 */
    "Meskel",                 /* Meskerem 17 */
    "Gena",                   /* Tahsas   29 */
    "Temqet",                  /* Ter      11 */
    "Victory of Adwa",          /* Yekatit  23 */
    "Good Friday",             /* Miazia -Final Friday */
    "Easter Sunday",            /* Miazia -Final Sunday */
    "International Labour Day", /* May       1 */
    "Eth. Patriots Day",        /* Miazia   27 */
    "Downfall of the Dergue"    /* Genbot   20 */
  };

        public int[] GregorianDaysPerMonth ={31,28,31, 30,31,30,31,31,30,31,30,31};
public          boolean       isBogusEthiopicDate ( Day d )
```

```
        {

         if ( !( 1 <= d.date && d.date <= 30 )
              || !(  1 <= d.month && d.month <= 13 )
              || ( d.month ==  13 && d.date > 6 )
              || ( d.month ==  13 && d.date == 6 && !isEthiopicLeapYear(d.year) )
            )
           return true;


         return false;
        }
        public boolean isLeapYear (long year )
        {
                        return ( (year%4==0) && ( (year%100!=0) || (year%400==0) ) );
        }
        public boolean isBogusGregorianDate ( Day d )
        {
                if(isLeapYear(d.year))
                        GregorianDaysPerMonth[1] = 29;
                else
                        GregorianDaysPerMonth[1] = 28;
                if ( !(    1 <= d.month && d.month <= 12 )
            || !( 1 <= d.date  && d.date  <= GregorianDaysPerMonth[d.month-1] ))
        {
                        GregorianDaysPerMonth[1] = 28;
                        return true;
                }
                GregorianDaysPerMonth[1] = 28;
                return false;
        }
        public static  int quotient ( double x,double y )
        {
                return ( (int)floor(x/y) );
        }
        public static double floor (double x)
        {
                long front, back;
                front = (int) x;
                if (x > 0)
                        return ( (double)front );
                /* else x is negative ... */
                back = (long)(10*Math.abs(x) - 10*Math.abs(front));
                return  ( back < 5 ) ? (double)front : (double)(front-1);
        }
```

```
	public long GregorianToFixed (Day d)
	{
		if ( isBogusGregorianDate ( d) )
		return (-1);
      return ( 365 * (d.year - 1L)
	 + quotient((d.year - 1L) , 4)
	 - quotient((d.year - 1L) , 100)
	 + quotient((d.year - 1L) , 400)
	 + quotient((367 * d.month - 362) , 12)
	 + (d.month <= 2 ? 0 : (isLeapYear(d.year) ? -1 : -2))
	 + d.date );
   }
	public int     GregorianToEthiopic (Day d)
	{
		long fixed;
		if ( isBogusGregorianDate ( d ) )
		return (-1);
		fixed = GregorianToFixed ( d);
		FixedToEthiopic ( fixed, d);
		return (1);
	}
	public void  FixedToEthiopic ( long fixed,Day d )
	{

	  d.year  = quotient ( (4 * (fixed - E_EPOCH) + 1463), 1461 );
	  Day temp =new Day();
	  temp.date = 1;
	  temp.month = 1;
	  temp.year = d.year;

	  d.month = quotient ( (fixed - EthiopicToFixed(temp)), 30 ) + 1;
	  temp.date = 1;
	  temp.month = d.month;
	  temp.year = d.year;

	  d.date  = (int)( fixed +1L - EthiopicToFixed(temp) );
	}
	public long  EthiopicToFixed (Day d )
	{
		if ( isBogusEthiopicDate ( d ) )
		return (-1);
		return ( E_EPOCH - 1 + 365 * (d.year - 1L) + quotient ( d.year, 4 )+ 30 * (d.month - 1)
+ d.date );
	}
	public boolean isEthiopicLeapYear(long eYear )
```

```
        {
                eYear += 1;
                return ( (eYear%4==0) );
        }
        public String getEthiopicDayOfWeek (Day d)
        {
          int day =(int) Math.abs( EthiopicToFixed ( d ) ) % 7;
          return ethDates [day];
        }
        public String getEthiopicMonth (int month)
        {
          return ethMonths [month-1];
        }
        public String getEthiopicYearName(long year)
        {
                return ethYearNames[((int)year)%4];
        }
        public int EthiopicLastDayOfMonth (int month,long year )
        {
          switch (month)
            {
              case 13:
                if ( isEthiopicLeapYear(year) )
                  return 6;
                else
                  return 5;
              default:
                return 30;
            }
        }
        public String isEthiopianHoliday (Day d )
        {
                //unsigned char** EthHolidays;
                //int *gDate, *gMonth;
                //long int *gYear;
                //unsigned char *returnString = NULL, *returnP = NULL;

                String returnP = null;



                //#ifdef WITHSERA
                  //if ( LCInfo & WITHSERA )
                    //EthHolidays = EthiopicSERAHolidays;
                  //else
        //#endif /* WITHSERA */
         // if (LCInfo & WITHUTF8)
           // EthHolidays = EthiopicUTF8Holidays;
          //else
           // EthHolidays = EthiopicTranscribedHolidays;


          switch ( d.month )
            {
              case 1:
                if ( d.date == 1 )              /* New Years */
                  returnP =  EthHolidays[0];
                else if ( d.date == 17 )
                  returnP =  EthHolidays[1];  /* Meskel    */
                break;

              case 4:
                if ( d.date == 29 )             /* Gena      */
                  returnP = EthHolidays[2];
                break;

              case 5:
                if ( d.date == 11 )             /* Temqet    */
                  returnP = EthHolidays[3];
                break;

              case 6:
                if ( d.date == 23 )             /* Adwa V-Day   */
                  returnP = EthHolidays[4];
                break;

              case 8:

                /* we need to check for easter here... */

                if ( d.date == 27 )            /* Patriots Day */
                  returnP = EthHolidays[8];
                break;

              case 9:
                if ( d.date == 20 )            /* Fall of Dergue */
                  returnP = EthHolidays[9];
                break;

              default:
                int n = EthiopicToGregorian (d);
                System.out.println(d.date+" "+d.month+" "+d.year);
                if (n!=-1 && d.date == 1 && d.month == 5 )  /* May Day/Labour Day */
                {
                    return ( EthHolidays[7] );
                }
                return ( null );
              }
            return ( returnP );

     }
     public int EthiopicToGregorian ( Day d )
     {
            long fixed;
            if ( isBogusEthiopicDate ( d ) )
            return (-1);
            fixed = EthiopicToFixed ( d );
            FixedToGregorian ( fixed, d );
            return (1);
     }
     public void FixedToGregorian (long fixed, Day d )
     {
            char correction;
            int priorDays;
            d.year  = yearFromFixed( fixed );
            Day temp = new Day();
            temp.year = d.year;
            temp.date = 1;
            temp.month = 1;
            priorDays = (int)( fixed - GregorianToFixed( temp ) );
            temp.year = d.year;
            temp.date = 1;
            temp.month = 3;
            if( fixed < GregorianToFixed( temp ) )
       correction =  0;
    else if(isLeapYear(d.year))
     correction = 1;
    else
     correction = 2;
            d.month = quotient ((12 * (priorDays + correction) + 373), 367);
            temp.year = d.year;
            temp.date = 1;
            temp.month = d.month;
            d.date = (int)(fixed - GregorianToFixed(temp ) + 1);
     }
```

```
public static long yearFromFixed ( long fixed )
{
        long d0 = fixed - 1L;           /* 1 is the Gregorian EPOCH */
        long n400 = quotient(d0, 146097);
        int  d1 = (int)(d0 % 146097);
        int  n100 = quotient(d1, 36524);
        int  d2 = d1 % 36524;
        int  n4 = quotient(d2 , 1461);
        int  d3 = d2 % 1461;
        int  n1 = quotient(d3, 365);
        /* int      d4 = d3 % 365 + 1; */
        long year = 400 * n400 + 100 * n100 + 4 * n4 + n1;
        return ( (n100 == 4 || n1 == 4) ? year : year + 1 );} }
```

Annexe E. Methods changed in Utils Class

```
 //Method Modified by Frezewd Lemma
// Changes are shaded.
        public static String getNiceDate(long time) {
       StringBuffer niceTime = new StringBuffer();
       GregorianCalendar cal = new GregorianCalendar();
       cal.setTime(new Date(time));

       Day d = new Day();
       d.date = new Integer(cal.get(Calendar.DAY_OF_MONTH)).intValue();
       d.month= new Integer(cal.get(Calendar.MONTH) + 1).intValue();
       d.year = new Long(cal.get(Calendar.YEAR)).longValue();
       Ethiopic e = new Ethiopic();
       e.GregorianToEthiopic(d);

       String day = "0" + d.date;
       String month = "0" + d.month;
       String year = new Long(d.year).toString();

       String hour = "0" + new Integer(cal.get(Calendar.HOUR) + 12
               * cal.get(Calendar.AM_PM)).intValue();
       String minute = "0" + new Integer(cal.get(Calendar.MINUTE));

       if(day.length() == 3) {
           day = day.substring(1, 3);
       }
       if(month.length() == 3) {
           month = month.substring(1, 3);
       }
       if(hour.length() == 3) {
           hour = hour.substring(1, 3);
       }
       if(minute.length() == 3) {
           minute = minute.substring(1, 3);
       }
       niceTime.append(day + ".");
       niceTime.append(month + ".");
       niceTime.append(year + " ");
       niceTime.append(hour + ":");
       niceTime.append(minute);
       return niceTime.toString();
   }

   /**
    * Gets a formated time string form a long time value.
```

```
     * @param time The time value as a long.
     * @return Formated time string.
     */

    public static String getNiceShortDate(CmsObject cms, long time) {
        StringBuffer niceTime = new StringBuffer();
        GregorianCalendar cal = new GregorianCalendar();
        cal.setTime(new Date(time));

       // CmsObject cms = new CmsObject();

        ExtendedProperties extendedProperties = null;

        // Collect the configurations
        try {
            extendedProperties = new
ExtendedProperties(CmsBase.getPropertiesPath(true));
        }
        catch(Exception e) {
            //throwInitException(new ServletException(C_ERRORMSG + "Trouble
reading property file " + CmsBase.getPropertiesPath(true) + ".\n\n", e));
        }

       Configurations conf = new Configurations(extendedProperties);

       try {
            // init the rb via the manager with the configuration
            // and init the cms-object with the rb.
            rb = CmsRbManager.init(conf);
        } catch(Exception e) {
            //if(C_LOGGING && isLogging(C_OPENCMS_CRITICAL))
log(C_OPENCMS_CRITICAL, ". Critical init error/3: " + e.getMessage());
            // any exception here is fatal and will cause a stop in processing
            //throw new CmsException("Database init failed",
CmsException.C_RB_INIT_ERROR, e);
        }


        String lang = CmsXmlLanguageFile.getCurrentUserLanguage(cms);
        String day="";
        String month="";
        String year="";

          if(lang.equals("am"))
          {
```

```
                    Day d = new Day();
          d.date = new Integer(cal.get(Calendar.DAY_OF_MONTH)).intValue();
          d.month= new Integer(cal.get(Calendar.MONTH) + 1).intValue();
          d.year = new Long(cal.get(Calendar.YEAR)).longValue();
          Ethiopic e = new Ethiopic();
          e.GregorianToEthiopic(d);

          day = "0" + d.date;
          month = "0" + d.month;
          year = new Long(d.year).toString();
          }
          else
          {
                    day            =            "0"            +            new
Integer(cal.get(Calendar.DAY_OF_MONTH)).intValue();
          month = "0" + new Integer(cal.get(Calendar.MONTH) + 1).intValue();
          year = new Integer(cal.get(Calendar.YEAR)).toString();
          }

        if(day.length() == 3) {
            day = day.substring(1, 3);
        }
        if(month.length() == 3) {
            month = month.substring(1, 3);
        }
        niceTime.append(day + ".");
        niceTime.append(month + ".");
        niceTime.append(year);
        return niceTime.toString();
    }
```

Annexe F. The Opencms_timer.js java script listing

```
// ===============================================
// JAVASCRIPT-FUNCTIONEN OPENCMS Mo
//
// dynamic timer
//
// author:          m.schleich
// company:         mindfact interaktive medien ag
// date:            24.01.2000
// update:
// update author:   Frezewd Lemma
//Every Modification point is shaded
// ===============================================



// global varibles and objects
var mday=0;
var E_EPOCH = 2796;
var weekday="";

// get date
   aktDat = new Date;
   aktTag= aktDat.getDate();
   aktMonat= aktDat.getMonth()+1;
   aktJahr= aktDat.getFullYear();

//get Ethiopian Date

        var yalenbetken=0;
        var yalenbetwor=0;
        var yalenbetamet=0;
        GregorianToEthiopic();


//to remeber the Date, which is slected by user
   userDat = new Date;
   userDay= userDat.getDate();
   userMonth= userDat.getMonth()+1;
   userYear= userDat.getFullYear();
   wday = 0;



//Ethiopic Functions
```

```
function isLeapYear ()
{
                return ( (aktJahr%4==0) && ( (aktJahr%100!=0) || (aktJahr%400==0) ) );
}



function isEthiopicLeapYear(eYear)
{
        eYear += 1;
        return ( (eYear%4==0) );
}

function quotient(x,y)
{
        z=x/y;
        msg=floor1(z);
        msg1=msg+"";
        r=parseInt(msg1);
        return (r);
}

function floor1(x)
{
        //long front, back;
        front1 = x + "";
        front=parseInt(front1);

        if (x > 0)
                {
                        f=front+"";
                        ff=parseFloat(f);
                        return (ff);
                }

        /* else x is negative ... */
        //back = (long)(10*Math.abs(x) - 10*Math.abs(front));
        back1 = (10*Math.abs(x) - 10*Math.abs(front)) + "";
        back2=parseInt(back1);
        fronts = front+"";
        f1=parseFloat(fronts);
        f2=parseFloat(fronts);
        f3=f2-1;
```

```
                if(back2 < 5) return(f1);else return (f3);

        }


        function GregorianTofixed ()
        {
                //if ( isBogusGregorianDate ( d) )
                //return (-1);
      return ( 365 * (aktJahr- 1)
         + quotient((aktJahr - 1) , 4)
         - quotient((aktJahr - 1) , 100)
         + quotient((aktJahr - 1) , 400)
         + quotient((367 * aktMonat - 362) , 12)
         + (aktMonat <= 2 ? 0 : (isLeapYear() ? -1 : -2))
         + aktTag );
   }



        function EthiopicTofixed (d)
        {
                //if ( isBogusEthiopicDate (d) )
                //return (-1);
         return ( E_EPOCH - 1 + 365 * (d.getFullYear() - 1) +
quotient( d.getFullYear(), 4 )+ 30 * (d.getMonth() - 1)+ d.getDate());
        }

//new


        function fixedToEthiopic (fixed1)
        {

          yalenbetamet= quotient((4 * (fixed1 - E_EPOCH) + 1463), 1461 );
          //Day temp =new Day();
          temp = new Date;
          temp.setDate(1);
          temp.setMonth(1);
          temp.setFullYear(yalenbetamet);

          yalenbetwor= quotient((fixed1 - EthiopicTofixed(temp)), 30 ) + 1;
          temp.setDate(1);
          temp.setMonth(yalenbetwor);
          temp.setFullYear(yalenbetamet);
          y = ( fixed1 +1 - EthiopicTofixed(temp) )+"";
          yalenbetken=parseInt(y);
        }

        function getEthiopicDayOfWeek(d)
        {

           day2=Math.abs(EthiopicTofixed(d))+"";
           day=parseInt(day2)%7;
           //day1=day+"";
           //da=parseInt(day1)
           return(day);
        }

        function GregorianToEthiopic ()
        {
                //long fixed;
                //if ( isBogusGregorianDate ( d ) )
                //return (-1);
                fixed1 = GregorianTofixed();
                fixedToEthiopic (fixed1);
                //return (1);
        }




// Object for month entry
function MonthEntry(sText, sValue)
  {
     this.sText = sText;
     this.sValue = sValue;
  }
// Object for year entry
function YearEntry(sText, sValue)
  {
     this.sText = sText;
     this.sValue = sValue;
  }
```

```
// arrays
var aYear = new Array();
function fillYear()
{
for(var j=0; j<11; j++)
{
   if(aWeekday[0]=="Mo")
        {
                help=aktJahr+j;
        }
        else
        {
                help=yalenbetamet+j;
        }
        aYear[j] = new YearEntry(help,help);
}
}

txt1='<tr><td align=right>1</td><td align=right>2</td><td align=right>3</td><td align
=right>4</td><td align=right>5</td><td align=right>6</td><td align=right>7</td></tr>';
txt12='<tr><td align=right>8</td><td align=right>9</td><td align=right>10</td><td align
=right>11</td><td align=right>12</td><td align=right>13</td><td align=right>14</td></tr>';
txt13='<tr><td align=right>15</td><td align=right>16</td><td align=right>17</td><td
 align=right>18</td><td align=right>19</td><td align=right>20</td><td align=right>21</td></t
txt14='<tr><td align=right>22</td><td align=right>23</td><td align=right>24</td><td
align=right>25</td><td align=right>26</td><td align=right>27</td><td align=right>28</td></tr
txt15a='<tr><td align=right>29</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td></tr>';
txt15b='<tr><td align=right>29</td><td align=right>30</td><td align=right>&nbsp;</td><td alig
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td></tr>';
txt15c='<tr><td align=right>29</td><td align=right>30</td><td align=right>31</td><td align=ri
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td></tr>';


p15='<tr><td align=right>1</td><td align=right>2</td><td align=right>3</td><td align=right>4<
align=right>&nbsp;</td><td align=right>&nbsp;</td></tr>';
p16='<tr><td align=right>1</td><td align=right>2</td><td align=right>3</td><td align=right>4<
align=right>6</td><td align=right>&nbsp;</td></tr>';




txt2='<tr><td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=rig
align=right>5</td><td align=right>6</td></tr>';
```

```
txt22='<tr><td align=right>7</td><td align=right>8</td><td align=right>9</td><td
align=right>10</td><td align=right>11</td><td align=right>12</td><td align=right>13
</td></tr>';
txt23='<tr><td align=right>14</td><td align=right>15</td><td align=right>16</td><td align=
righ\t>17</td><td align=right>18</td><td align=right>19</td><td align=right>20</td></tr>';
txt24='<tr><td align=right>21</td><td align=right>22</td><td align=right>23</td><td align=
right>24</td><td align=right>25</td><td align=right>26</td><td align=right>27</td></tr>';
txt25a='<tr><td align=right>28</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td
 align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td></tr>';
txt25b='<tr><td align=right>28</td><td align=right>29</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td></tr>';
txt25c='<tr><td align=right>28</td><td align=right>29</td><td align=right>30</td><td align=
right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';
txt25d='<tr><td align=right>28</td><td align=right>29</td><td align=right>30</td><td align=
right>31</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';

p25='<tr><td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=
right>3</td><td align=right>4</td><td align=right>5</td><td align=right>&nbsp;</td></tr>';
p26='<tr><td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=
right>3</td><td align=right>4</td><td align=right>5</td><td align=right>6</td></tr>';


txt3='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td
align=right>2</td><td align=right>3</td><td align=right>4</td><td align=right>5</td></tr>';
txt32='<tr><td align=right>6</td><td align=right>7</td><td align=right>8</td><td align=right>
9</td><td align=right>10</td><td align=right>11</td><td align=right>12</td></tr>';
txt33='<tr><td align=right>13</td><td align=right>14</td><td align=right>15</td><td align=
right>16</td><td align=right>17</td><td align=right>18</td><td align=right>19</td></tr>';
txt34='<tr><td align=right>20</td><td align=right>21</td><td align=right>22</td><td align=
right>23</td><td align=right>24</td><td align=right>25</td><td align=right>26</td></tr>';
txt35a='<tr><td align=right>27</td><td align=right>28</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td></tr>';
txt35b='<tr><td align=right>27</td><td align=right>28</td><td align=right>29</td><td align=
right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';
txt35c='<tr><td align=right>27</td><td align=right>28</td><td align=right>29</td><td align=
right>30</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';
txt35d='<tr><td align=right>27</td><td align=right>28</td><td align=right>29</td><td align
```

```
=right>30</td><td align=right>31</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';


p35='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td
align=right>2</td><td align=right>3</td><td align=right>4</td><td align=right>5</td></tr>';
p36='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td
align=right>2</td><td align=right>3</td><td align=right>4</td><td align=right>5</td></tr>';
p362='<tr><td align=right>6</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td></tr>';



txt4='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>1</td><td align=right>2</td><td align=right>3</td><td align=right>4</td>
</tr>';
txt42='<tr><td align=right>5</td><td align=right>6</td><td align=right>7</td><td align=right>
8</td><td align=right>9</td><td align=right>10</td><td align=right>11</td></tr>';
txt43='<tr><td align=right>12</td><td align=right>13</td><td align=right>14</td><td align=
right>15</td><td align=right>16</td><td align=right>17</td><td align=right>18</td></tr>';
txt44='<tr><td align=right>19</td><td align=right>20</td><td align=right>21</td><td align=
right>22</td><td align=right>23</td><td align=right>24</td><td align=right>25</td></tr>';
txt45a='<tr><td align=right>26</td><td align=right>27</td><td align=right>28</td><td align=
right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';
txt45b='<tr><td align=right>26</td><td align=right>27</td><td align=right>28</td><td align=
right>29</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp
;</td></tr>';
txt45c='<tr><td align=right>26</td><td align=right>27</td><td align=right>28</td><td align=
right>29</td><td align=right>30</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
</tr>';
txt45d='<tr><td align=right>26</td><td align=right>27</td><td align=right>28</td><td align=
right>29</td><td align=right>30</td><td align=right>31</td><td align=right>&nbsp;</td></tr>
';



p45='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>1</td><td align=right>2</td><td align=right>3</td><td align=right>4</td></tr
>';
p452='<tr><td align=right>5</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right
>&nbsp;</td></tr>';
p46='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
```

```
<td align=right>1</td><td align=right>2</td><td align=right>3</td><td align=right>4</td>
</tr>';
p462='<tr><td align=right>5</td><td align=right>6</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';




txt5='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=right>3
</td></tr>';
txt52='<tr><td align=right>4</td><td align=right>5</td><td align=right>6</td><td align=right>
6</td><td align=right>8</td><td align=right>9</td><td align=right>10</td></tr>';
txt53='<tr><td align=right>11</td><td align=right>12</td><td align=right>13</td><td align=
right>14</td><td align=right>15</td><td align=right>16</td><td align=right>17</td></tr>';
txt54='<tr><td align=right>18</td><td align=right>19</td><td align=right>20</td><td align=
right>21</td><td align=right>22</td><td align=right>23</td><td align=right>24</td></tr>';
txt55a='<tr><td align=right>25</td><td align=right>26</td><td align=right>27</td><td align=
right>28</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';
txt55b='<tr><td align=right>25</td><td align=right>26</td><td align=right>27</td><td align=
right>28</td><td align=right>29</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
</tr>';
txt55c='<tr><td align=right>25</td><td align=right>26</td><td align=right>27</td><td align=
right>28</td><td align=right>29</td><td align=right>30</td><td align=right>&nbsp;</td>
</tr>';
txt55d='<tr><td align=right>25</td><td align=right>26</td><td align=right>27</td><td align=
right>28</td><td align=right>29</td><td align=right>30</td><td align=right>31</td></tr>';



p55='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=right>3
</td></tr>';
p552='<tr><td align=right>4</td><td align=right>5</td><td align=right>&nbsp;</td><td align=
right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';
p56='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>1</td><td align=right>2</td><td align=right>3
</td></tr>';
p562='<tr><td align=right>4</td><td align=right>5</td><td align=right>6</td><td align=right>
&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';
```

```
txt6='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td align=
right>2</td></tr>';
txt62='<tr><td align=right>3</td><td align=right>4</td><td align=right>5</td><td align=right>
6</td><td align=right>7</td><td align=right>8</td><td align=right>9</td></tr>';
txt63='<tr><td align=right>10</td><td align=right>11</td><td align=right>12</td><td align=
right>13</td><td align=right>14</td><td align=right>15</td><td align=right>16</td></tr>';
txt64='<tr><td align=right>17</td><td align=right>18</td><td align=right>19</td><td align=
right>20</td><td align=right>21</td><td align=right>22</td><td align=right>23</td></tr>';
txt65a='<tr><td align=right>24</td><td align=right>25</td><td align=right>26</td><td align=
right>27</td><td align=right>28</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
</tr>';
txt65b='<tr><td align=right>24</td><td align=right>25</td><td align=right>26</td><td align=
right>27</td><td align=right>28</td><td align=right>29</td><td align=right>&nbsp;</td>
</tr>';
txt65c='<tr><td align=right>24</td><td align=right>25</td><td align=right>26</td><td align=
right>27</td><td align=right>28</td><td align=right>29</td><td align=right>30</td></tr>';
txt66='<tr><td align=right>31</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td></tr>';



p65='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td align=right>
2</td></tr>';
p652='<tr><td align=right>3</td><td align=right>4</td><td align=right>5</td><td align=right>
&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';
p66='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>1</td><td
align=right>2</td></tr>';
p662='<tr><td align=right>3</td><td align=right>4</td><td align=right>5</td><td align=
right>6</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';



txt7='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td><td align=right>1</td></tr>';
txt72='<tr><td align=right>2</td><td align=right>3</td><td align=right>4</td><td align=
right>5</td><td align=right>6</td><td align=right>7</td><td align=right>8</td></tr>';
txt73='<tr><td align=right>9</td><td align=right>10</td><td align=right>11</td><td
align=right>12</td><td align=right>13</td><td align=right>14</td><td align=right>15
</td></tr>';
txt74='<tr><td align=right>16</td><td align=right>17</td><td align=right>18</td><td
align=right>19</td><td align=right>20</td><td align=right>21</td><td align=right>22
</td></tr>';
txt75a='<tr><td align=right>23</td><td align=right>24</td><td align=right>25</td><td
align=right>26</td><td align=right>27</td><td align=right>28</td><td align=right>&nbsp
;</td></tr>';
txt75b='<tr><td align=right>23</td><td align=right>24</td><td align=right>25</td><td
align=right>26</td><td align=right>27</td><td align=right>28</td><td align=right>29</td>
</tr>';
txt76a='<tr><td align=right>30</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td></tr>';
txt76b='<tr><td align=right>30</td><td align=right>31</td><td align=right>&nbsp;</td>
<td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td
align=right>&nbsp;</td></tr>';




p75='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td>
<td align=right>1</td></tr>';
p752='<tr><td align=right>2</td><td align=right>3</td><td align=right>4</td><td align=
right>5</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>
&nbsp;</td></tr>';
p76='<tr><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp
;</td><td align=right>&nbsp;</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td><td align=right>1</td></tr>';
p762='<tr><td align=right>2</td><td align=right>3</td><td align=right>4</td><td align=
right>5</td><td align=right>6</td><td align=right>&nbsp;</td><td align=right>&nbsp;
</td></tr>';




// generates two selctboxes for year and month
function writeSel() // fill the year and month selectbox
  {
     weekday=aWeekday[0];
        var SelHtml = '';
     var MoHtml = '';
     if(ns)
     {
        SelHtml = '<select name="YEAR" width="50" STYLE="WIDTH:50px">';
        SelHtml += '<select name="YEAR" width="50" STYLE="WIDTH:50px"
onChange="selectDays(1,TIMER.MONTH.options[TIMER.MONTH.selectedIndex].value,this.options
[this.selectedIndex].value);">';
     }
     else
     {
       SelHtml = '<select name="YEAR" width="100" STYLE="WIDTH:50px"
onChange="selectDays(1,TIMER.MONTH.options[TIMER.MONTH.selectedIndex].value,this.options
[this.selectedIndex].value);">';
     }
     for(var j=0; j<11; j++)
     {
       SelHtml += '<OPTION VALUE=""></OPTION>';
     }
     SelHtml += '</SELECT>';


     MoHtml = '</td><td><select name="MONTH" width="100" STYLE="WIDTH:100px"
onChange="selectDays(1,this.options[this.selectedIndex].value,TIMER.YEAR.options
[TIMER.YEAR.selectedIndex].value);">';
//if(aWeekday[0]=="Mo")nm=13; else nm=12;
     for(var j=0; j<13; j++)
     {
          if(aWeekday[0]=="Mo")
               {
                         if((j+1)==aktMonat)
               {
                         MoHtml += '<OPTION VALUE="" selected></OPTION>';
               }
               else
               {
                         MoHtml += '<OPTION VALUE=""></OPTION>';
               }
               }
               else
               {

                         if((j+1)==yalenbetwor)
                         {
                                 MoHtml += '<OPTION VALUE="" selected></OPTION>';
                         }
                         else
```

```
                {
                        MoHtml += '<OPTION VALUE=""></OPTION>';
                }

                }

      }
      MoHtml += '</SELECT></td>';


     SelHtml=  new String( SelHtml+MoHtml );

     document.open();
     document.clear();
     document.write(SelHtml);
     document.close();
  }

// fills the two selctboxes for year and month
function fillBox(objSel, array, length)
{
   for(i=0; i<length; i++)
      {
         objSel.options[i].text = array[i].sText;
         objSel.options[i].value = array[i].sValue;
      }
}



// decides which layer with days is shown
function selectDays(day, month, year)
{

   mday = checkTimerDate(month, year);

   userDat.setFullYear(year);
   if(aWeekday[0]=="Mo")userDat.setMonth(month-1);
   else userDat.setMonth(month);

   userDat.setDate(1);



   if(aWeekday[0]=="Mo")
   {

                  wday = userDat.getUTCDay();
  }
   else
  {
                  wday =getEthiopicDayOfWeek(userDat);

  }



   if (wday==0){wday=7;}    //sunday is 0 in Javasicrtpt

   hidelyr('days');

if(aWeekday[0]=="Mo")
{
        switch (mday)
      {
          case 31:
          {
              switch (wday)
              {
                  case 1:
                  {
                      txt=txt1+txt12+txt13+txt14+txt15c;
                      break;
                  }
                  case 2:
                  {
                      txt=txt2+txt22+txt23+txt24+txt25d;
                      break;
                  }
                  case 3:
                  {
                      txt=txt3+txt32+txt33+txt34+txt35d;
                      break;
                  }
                  case 4:
                  {
                      txt=txt4+txt42+txt43+txt44+txt45d;
                      break;
                  }
                  case 5:
```

```
          {
            txt=txt5+txt52+txt53+txt54+txt55d;
                    break;
          }
          case 6:
          {
             txt=txt6+txt62+txt63+txt64+txt65c+txt66;
                     break;
          }
          case 7:
          {
            txt=txt7+txt72+txt73+txt74+txt75b+txt76b;
            break;
          }
        }
        break;
      }
      case 30:
      {
        switch (wday)
        {
          case 1:
          {
            txt=txt1+txt12+txt13+txt14+txt15b;
            break;
          }
          case 2:
          {
            txt=txt2+txt22+txt23+txt24+txt25c;
            break;
          }
          case 3:
          {
            txt=txt3+txt32+txt33+txt34+txt35c;
            break;
          }
          case 4:
          {
            txt=txt4+txt42+txt43+txt44+txt45c;
            break;
          }
          case 5:
          {
            txt=txt5+txt52+txt53+txt54+txt55c;
            break;
          }
          case 6:
          {
             txt=txt6+txt62+txt63+txt64+txt65c;
             break;
          }
          case 7:
          {
             txt=txt7+txt72+txt73+txt74+txt75b+txt76a;
             break;
          }
       }
       break;
    }
    case 29:
    {
       switch (wday)
       {
          case 1:
          {
             txt=txt1+txt12+txt13+txt14+txt15a;
             break;
          }
          case 2:
          {
             txt=txt2+txt22+txt23+txt24+txt25b;
             break;
          }
          case 3:
          {
             txt=txt3+txt32+txt33+txt34+txt35b;
             break;
          }
          case 4:
          {
             txt=txt4+txt42+txt43+txt44+txt45b;
             break;
          }
          case 5:
          {
             txt=txt5+txt52+txt53+txt54+txt55b;
             break;
          }
          case 6:
          {
                txt=txt6+txt62+txt63+txt64+txt65b;
                break;
            }
            case 7:
            {
                txt=txt7+txt72+txt73+txt74+txt75b;
                break;
            }
        }
        break;
    }
    case 28:
    {
        switch (wday)
        {
            case 1:
            {
                txt=txt1+txt12+txt13+txt14;
                break;
            }
            case 2:
            {
                txt=txt2+txt22+txt23+txt24+txt25a;
                break;
            }
            case 3:
            {
                txt=txt3+txt32+txt33+txt34+txt35a;
                break;
            }
            case 4:
            {
                txt=txt4+txt42+txt43+txt44+txt45a;
                break;
            }
            case 5:
            {
                txt=txt5+txt52+txt53+txt54+txt55a;
                break;
            }
            case 6:
            {
                txt=txt6+txt62+txt63+txt64+txt65a;
                break;
            }
```

```
                case 7:
                {
                    txt=txt7+txt72+txt73+txt74+txt75a;
                    break;
                }
            }
            break;
        }
    }

}
else
{
        switch(mday)
        {

                case 30:
                        {
             switch (wday)
                {
                case 1:
                        {
                        txt=txt1+txt12+txt13+txt14+txt15b;
                        break;
                        }
                 case 2:
                        {
                        txt=txt2+txt22+txt23+txt24+txt25c;
                        break;
                        }
                        case 3:
                        {
                        txt=txt3+txt32+txt33+txt34+txt35c;
                        break;
                        }
                        case 4:
                        {
                        txt=txt4+txt42+txt43+txt44+txt45c;
                        break;
                        }
                        case 5:
                        {
                        txt=txt5+txt52+txt53+txt54+txt55c;
                        break;
                        }
                    case 6:
                    {
                    txt=txt6+txt62+txt63+txt64+txt65c;
                    break;
                    }
                    case 7:
                    {
                    txt=txt7+txt72+txt73+txt74+txt75b+txt76a;
                    break;
                    }
                              default:
                              {
                                        txt=p15+p16;
                                        break;
                              }
          }
          break;
             }
             case 5:
                    {

                               switch (wday)
               {
              case 1:
                    {
                    txt=p15;
                    break;
                    }
                    case 2:
                    {
                    txt=p25;
                    break;
                    }
                    case 3:
                    {
                    txt=p35;
                    break;
                    }
                    case 4:
                    {
                    txt=p45+p452
                    break;
                    }
                    case 5:
                    {
```

```
                txt=p55+p552;
                break;
                }
                case 6:
                {
                txt=p65+p652;
                break;
                }
                case 7:
                {
                txt=p75+p752;
                break;
                }
                        default:
                        {
                                txt=p15+p16;
                                break;
                        }
        }

     break;
                }
        case 6:
                {
         switch (wday)
        {
                case 1:
                {
                txt=p16;
                break;
                }
                case 2:
                {
                txt=p26;
                break;
                }
                case 3:
                {
                txt=p36+p362;
                    break;
                }
                case 4:
                {
                txt=p46+p462;
                break;
```

```
                    }
                    case 5:
                    {
                    txt=p56+p562;
                    break;
                    }
                    case 6:
                    {
                    txt=p66+p662;
                    break;
                    }
                    case 7:
                    {
                    txt=p76+p762;
                    break;
                    }
                         default:
                         {
                              txt=p15+p16;
                              break;
                         }
               }
             break;
                    }
     default:
     {
          txt=p15+p16;
          break;
     }
  }
}
  txt = '<table border=0 cellspacing=1 cellpadding=4><tr><td align=center>'+aWeekday[0]+
'</td><td align=center>'+aWeekday[1]+'</td><td align=center>'+aWeekday[2]+'</td><td
align=center>'+aWeekday[3]+'</td><td align=center>'+aWeekday[4]+'</td><td align=center>
'+aWeekday[5]+'</td><td align=center>'+aWeekday[6]+'</td></tr><tr><td colspan=7><hr>
</td></tr>'+txt+'</table>';
  eval(lyrtxt);
  eval(layerzeigen_01+'days'+layerzeigen_02);
  userYear=year;
  if(month<10)
  {
    userMonth='0'+month;
  }
  else
  {
        userMonth=month;
    }
    changeDay(wday);

}

//select amharic date or english
function sel()
{
          if(aWeekday[0]=="Mo")selectDays(aktTag, aktMonat, aktJahr);
          else selectDays(yalenbetken,yalenbetwor,yalenbetamet);
}



// changes the selected day
function changeDay(day)
{
    hilf=wday-1;
    hilf2=day-1;


    if(day>=wday && day<(mday+wday))
    {
        for(i=0; i<37; i++)
        {
            eval(imgonlyr+'images[i].src = "' + resourcesUri + 'empty.gif"');
        }
        eval(imgonlyr+'images[hilf2].src = "' + resourcesUri + 'circle.gif"');
        if((day-hilf)<10)
            {
                userDay='0'+new String(day-hilf);
            }
        else
        {
            userDay=day-hilf;
        }

    }
}




//-------------------------------------------------------------------------------
// ceck days
//
```

```
// author:  Michaela Schleich
// company: mindfact interaktive medien ag
// date:    25.01.2000
// update:
//
// input date (int month, int year)
//
// returns days per month &#x1325;&#x122d
//--------------------------------------------------------------------------------
function checkTimerDate(month, year)
{
  var schaltjahr=false;
      if(aWeekday[0]=="Mo")
      {
           if(month == 2)
         {
           sj = year%4;
           if(sj==0)
           schaltjahr=true;

           sj = year%100;
           if(sj==0)
           schaltjahr=false;

           sj = year%400;
           if(sj==0)
           schaltjahr=true;
         }

         if(schaltjahr)
         {
           return 29;
         }
         else
         {
           if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 ||
month==12)
           {
             return 31;
           }
           else
           {
             if(month == 2)
             {
               return 28;
```

```
                }
                else
                {
                    return 30;
                }
            }
        }

    }

    else
    {
            if(month==13)
            {
                    //Here calculate Ethiopian Leap year
                    if(isEthiopicLeapYear(year))
                    schaltjahr=true;

                    if(schaltjahr)
                    {
                            return 6;
                    }
                    else return 5;
            }
            else
                    return 30;
    }

}
```

Annexe G. The Timer template listing

```
<?xml version="1.0" encoding="UTF-8"?>
<WORKPLACE>
<TEMPLATE>
<![CDATA[

<html>
<head>
<!-- Beginn Style -->
<link rel=stylesheet type="text/css" href="]]><method
name="resourcesUri">format.css</method><![CDATA[">
<!-- Ende Style -->

<!-- Beginn externe Scripte wegen Datumspruefung-->

<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript" SRC="]]><method
name="scriptsUri">opencms_choosebrowser.js</method><![CDATA["></SCRIPT>
<script language="JavaScript">
   var resourcesUri="]]><METHOD name="resourcesUri"/><![CDATA[";
</script>
<!-- fuer Kalender -->

<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript" SRC="]]><method
name="scriptsUri">opencms_timer.js</method><![CDATA["></SCRIPT>


<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">

var aMonth = new Array();
var aMonthe=new Array();

aMonth[0] = new MonthEntry(']]><LABEL value="calendar.months.jan" /><![CDATA[',1);
aMonth[1] = new MonthEntry(']]><LABEL value="calendar.months.feb" /><![CDATA[',2);
aMonth[2] = new MonthEntry(']]><LABEL value="calendar.months.mar" /><![CDATA[',3);
aMonth[3] = new MonthEntry(']]><LABEL value="calendar.months.apr" /><![CDATA[',4);
aMonth[4] = new MonthEntry(']]><LABEL value="calendar.months.may" /><![CDATA[',5);
aMonth[5] = new MonthEntry(']]><LABEL value="calendar.months.jun" /><![CDATA[',6);
aMonth[6] = new MonthEntry(']]><LABEL value="calendar.months.jul" /><![CDATA[',7);
aMonth[7] = new MonthEntry(']]><LABEL value="calendar.months.aug" /><![CDATA[',8);
aMonth[8] = new MonthEntry(']]><LABEL value="calendar.months.sep" /><![CDATA[',9);
aMonth[9] = new MonthEntry(']]><LABEL value="calendar.months.okt" /><![CDATA[',10);
aMonth[10] = new MonthEntry(']]><LABEL value="calendar.months.nov" /><![CDATA[',11);
aMonth[11] = new MonthEntry(']]><LABEL value="calendar.months.dec" /><![CDATA[',12);
aMonthe[0] = new MonthEntry(']]><LABEL value="calendar.months.sep" /><![CDATA[',1);
aMonthe[1] = new MonthEntry(']]><LABEL value="calendar.months.okt" /><![CDATA[',2);
aMonthe[2] = new MonthEntry(']]><LABEL value="calendar.months.nov" /><![CDATA[',3);
aMonthe[3] = new MonthEntry(']]><LABEL value="calendar.months.dec" /><![CDATA[',4);
aMonthe[4] = new MonthEntry(']]><LABEL value="calendar.months.jan" /><![CDATA[',5);
aMonthe[5] = new MonthEntry(']]><LABEL value="calendar.months.feb" /><![CDATA[',6);
aMonthe[6] = new MonthEntry(']]><LABEL value="calendar.months.mar" /><![CDATA[',7);
aMonthe[7] = new MonthEntry(']]><LABEL value="calendar.months.apr" /><![CDATA[',8);
aMonthe[8] = new MonthEntry(']]><LABEL value="calendar.months.may" /><![CDATA[',9);
aMonthe[9] = new MonthEntry(']]><LABEL value="calendar.months.jun" /><![CDATA[',10);
aMonthe[10] = new MonthEntry(']]><LABEL value="calendar.months.jul" /><![CDATA[',11);
aMonthe[11] = new MonthEntry(']]><LABEL value="calendar.months.aug"
/><![CDATA[',12);
aMonthe[12] = new MonthEntry(']]><LABEL value="calendar.months.pag"
/><![CDATA[',13);

var aWeekday = new Array();

aWeekday[0] = ']]><LABEL value="calendar.weekday.mo" /><![CDATA[';
aWeekday[1] = ']]><LABEL value="calendar.weekday.tu" /><![CDATA[';
aWeekday[2] = ']]><LABEL value="calendar.weekday.we" /><![CDATA[';
aWeekday[3] = ']]><LABEL value="calendar.weekday.th" /><![CDATA[';
aWeekday[4] = ']]><LABEL value="calendar.weekday.fr" /><![CDATA[';
aWeekday[5] = ']]><LABEL value="calendar.weekday.sa" /><![CDATA[';
aWeekday[6] = ']]><LABEL value="calendar.weekday.su" /><![CDATA[';

function updateCal() {  //sets calender field to selected date
   var toDo = "window.opener.top.body."+ window.opener.parent.frames[1].name;
   toDo +=
".document.NEWTASK.DATE.value=userDay+'.'+userMonth+'.'+userYear;window.close();"
   eval(toDo);
}


</SCRIPT>


<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=]]><METHOD
name="getEncoding"/><![CDATA[">

<title>]]><LABEL value="calendar.title" /><![CDATA[</title>
</head>

<body bgcolor="#ffffff" background="]]><METHOD
```

```
name="resourcesUri">bg_weiss.gif</METHOD><![CDATA["  bgproperties=fixed
onLoad="sel();">

<form id="TIMER" name="TIMER">
<table border=0 cellspacing=0 cellpadding=5>
<tr>
    <td align=right>
        <SCRIPT LANGUAGE="JavaScript">
            writeSel();
        </SCRIPT>
    </td>
</tr>
</table>
</form>

<div id="top" style="FONT-FAMILY: MS Sans Serif, Arial, helvetica, sans-serif; FONT-SIZE:
8pt; position: absolute; LEFT: 50px; TOP: 93px; VISIBILITY: schown; Z-INDEX: 100;">
<table border=0 cellspacing=1 cellpadding=0>

    <tr>
        <td align=center><a href="javascript:changeDay(1)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="1" border=0 width=20
height=21></td>
        <td align=center><a href="javascript:changeDay(2)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="2" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(3)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="3" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(4)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="4" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(5)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="5" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(6)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="6" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(7)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="7" border=0 width=20
height=21></a></td>
    </tr>
    <tr>
        <td align=center><a href="javascript:changeDay(8)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="8" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(9)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="9" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(10)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="10" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(11)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="11" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(12)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="12" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(13)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="13" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(14)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="14" border=0 width=20
height=21></a></td>
    </tr>
    <tr>
        <td align=center><a href="javascript:changeDay(15)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="15" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(16)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="16" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(17)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="17" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(18)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="18" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(19)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="19" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(20)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="20" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(21)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="21" border=0 width=20
height=21></a></td>
    </tr>
    <tr>
        <td align=center><a href="javascript:changeDay(22)"><img src="]]><method
```

```
name="resourcesUri">empty.gif</method><![CDATA[" name="22" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(23)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="23" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(24)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="24" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(25)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="25" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(26)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="26" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(27)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="27" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(28)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="28" border=0 width=20
height=21></a></td>
    </tr>
    <tr>
        <td align=center><a href="javascript:changeDay(29)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="29" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(30)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="30" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(31)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="31" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(32)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="32" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(33)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="33" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(34)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="34" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(35)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="35" border=0 width=20
height=21></a></td>
    </tr>
    <tr>
```

```
        <td align=center><a href="javascript:changeDay(36)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="36" border=0 width=20
height=21></a></td>
        <td align=center><a href="javascript:changeDay(37)"><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" name="37" border=0 width=20
height=21></a></td>
        <td align=center colspan=5>&nbsp;</td>
    </tr>
</table>
</div>

<!---------------------------------------------->
<!-- Layer with timer day sheet                -->
<!---------------------------------------------->
<div id="days" style="FONT-FAMILY: MS Sans Serif, Arial, helvetica, sans-serif; FONT-SIZE:
8pt; position: absolute; LEFT: 45px; TOP: 45px; VISIBILITY: hidden; Z-INDEX: 0;">
</div>
<!-- Buttons              -->
<form>
<table border=0 cellspacing=0 cellpadding=5>
<tr><td colspan=2><img src="]]><method
name="resourcesUri">empty.gif</method><![CDATA[" border=0 width=1
height=165></td></tr>
<tr>
    <td align=right width=50%><INPUT class=button width=100 type="button"
value="]]><LABEL value="button.submit" /><![CDATA[" id=OK name=OK
onClick="updateCal()"></td>
    <td align=left width=50%><INPUT class=button width=100 type="button"
value="]]><LABEL value="button.cancel" /><![CDATA[" id=CANCEL name=CANCEL
onClick="window.close();"></td>
</tr>
</table>
</form>

<SCRIPT LANGUAGE="JavaScript">
//aMonth[0]="&#x1325;&#x122d";

fillYear();
if(aWeekday[0]=="Mo")
{
          fillBox(document.TIMER.YEAR, aYear, 11);
          fillBox(document.TIMER.MONTH, aMonth, 12);
}
else
{
        fillBox(document.TIMER.YEAR, aYear, 11);
        fillBox(document.TIMER.MONTH, aMonthe, 13);

}
</SCRIPT>
</body>
</html>
]]>
</TEMPLATE>
</WORKPLACE>
```

Annexe H. Ant Installation

To install Ant, choose a directory and copy the distribution file there. This directory will be known as ANT_HOME.

Setup

Before you can run ant there is some additional set up you will need to do:

- Add the bin directory to your path.
- Set the ANT_HOME environment variable to the directory where you installed Ant. On some operating systems the ant wrapper scripts can guess ANT_HOME (Unix dialects and Windows NT/2000) - but it is better to not rely on this behavior.
- Optionally, set the JAVA_HOME environment variable.

  This should be set to the directory where your JDK is installed.

To build the modified source code with ant, it needs to work on the command prompt. After displaying the command window and changing the directory to c:\source\opencsource, issuing the command **ant all** finished the job of building opencms from the source code.

Annexe I. Test Class generated for the testing of the calendar program

```
class Test
{
    public static void main(String[] args)
    {
            Ethiopic g = new Ethiopic();
            Day d = new Day();
            //Here is where dates are changed for testing purpose
           d.date = 8;
            d.month = 9;
            d.year = 2004;
            //System.out.println(g.isLeapYear(d.year)+""+d.year);
            System.out.println("\nGregorian:
Date:"+d.date+"\tMonth:"+d.month+"\tYear:"+d.year);
            int c = g.GregorianToEthiopic(d);
            if(c==1)
            {
                    System.out.println("\n
Ethiopic:Date:"+d.date+"\tMonth:"+d.month+"\tYear:"+d.year);
                    System.out.println(g.getEthiopicYearName(d.year));
                    System.out.println(g.getEthiopicDayOfWeek(d)+"
"+g.getEthiopicMonth(d.month)+" "+d.date+","+d.year);
                    if(g.isEthiopianHoliday(d)!=null)
                            System.out.println(g.isEthiopianHoliday(d));
            }
}
```

Annexe J. Listing of Ethiopic.jsp

```
<%
Ethiopic e = new Ethiopic();
Day d = new Day();
%>
<p>Ethiopian Calendar</p>
<table border=2>
<tr bgcolor="#cccccc"><th colspan=7> <form  method=post action="" >
   &#x12C8;&#x122D;: <select name=month lang="am">
   <%
     Calendar c = Calendar.getInstance(  );

     d.year = c.get(Calendar.YEAR);
     d.month = c.get(Calendar.MONTH)+1;
     d.date = c.get(Calendar.DATE);
     e.GregorianToEthiopic(d);



//Displays the months in the drop down list box
     for (int i=0; i<e.ethUTFMonths.length; i++) {
     out.print("<option>");
     out.print(e.ethMonths[i]);
     out.println("</option>");
   }
   %>
   </select>
   &#x12D3;&#x1218;&#x1275; (&#x1263;&#x1208; 4-
&#x1241;&#x1325;&#x122D;):
     <input type="text" size="5" name="year"
       value="<%= d.year %>"></input>
   <input type=submit value="&#x12A0;&#x1233;&#x12ED">
 </form></tr>


<%
//Displays the days name on the table
     out.print("<tr>");
     for (int i=0; i<e.ethUTFDates.length; i++) {
      out.print("<td>");
      out.print(e.ethUTFDates[i]);
      out.print("</td>");
        }
     out.print("</tr>");
```

```
//Finds the leading gap
  Day temp = new Day();
  temp.date = 1;
  temp.month = d.month;
  temp.year = d.year;
  int leadGap = e. getEthiopicDayOfWeek(temp);

// Finds the number of days in a given month
 int daysInMonth = e.EthiopicLastDayOfMonth(d.month, d.year);

     out.print("<tr>");

     // Blank out the labels before 1st day of month
     for (int i = 0; i < leadGap; i++) {
       out.print("<td>&nbsp;");
     }

     // Fill in numbers for the day of month.
     for (int i = 1; i <= daysInMonth; i++) {

       out.print("<td>");
       out.print(i);
       out.print("<br/><font size=\"3\">");
       //sd = Integer.toString(i);
       //if (i <10) sd = "0" + sd;

              out.print("</font></td>");
       if ((leadGap + i) % 7 == 0) {    // wrap if end of line.
         out.println("</tr>");
         out.print("<tr>");
       }
}
%>
</table>
<%
out.print(d.month);
out.print("/");
out.print(d.date);
out.print("/");
out.print(d.year);
%>
</body>
</html>
```

Annexe K. Interview Questions

Interview 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 considered.

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. Therefore, electronic document management and exchange using multiple languages is the solution to these and other related problems.

At different times during the interview, the Interviewee has promised to give the following documents for research purpose.

- Documents prepared by UNDP consultancy
- Documents prepared by Microsoft – TOR document & document contains about data centers
- Document exchange needs by woredas done by Dr. Assefa
- Infrastructure solution document
- Different Documents available at Ato Zelalem & Ato Ghion