CS297 Report
Enhancing open source localization

Farzana Forhad

farzanaforhad2003@yahoo.com

Advisor: Dr. Chris Pollett

Department of Computer Science

San Jose State University

FALL 2008

Table of Contents

1.0 INTRODUCTION

Many non-English speakers are more comfortable browsing the web in their mother tongue rather than English. Most of the time, they like to see web pages in their own language, such as, Chinese, Japanese, Latin and so on. There are many softwares which have been used for translating the web pages. The particular kind of software which displays text in a user's local/native language is called localization (l10n) software.

On the other hand, internationalization is a process which is needed for the localization and deals with the more general problem of making software easy to localize to any language. Internationalization (i18n) is used to design the software application in such a way so that it could be used by any languages without doing any sort of engineering changes to it.

“gettext” is an important part of localization and internationalization. “gettext” is an open source tool for internationalization. Translating strings into different languages using “gettext” tools takes several steps. The goal of this project is to enhance “gettext”- so that it takes lesser steps for the Engineers to localize the text.

“gettext” requires several steps to create web pages that work in any languages. To translate English version of web pages to other languages requires compilation for each translation which is very time consuming and expensive. The enhancement to “gettext” is to eliminate the need for any compilation in producing English version of web pages yet still supporting unique identifiers for page items so that they can be easily localized to other languages.

Apart from localization the purpose of this project is also to enhance the dependency of database on localization.

In Deliverable 1 we used an existing example to learn how the “gettext” works. In Deliverable 2 we downloaded the source code for localizing string and made some small changes like converting every word to its pig-latinized word for any given language. For Deliverable 3 we set up database table where different strings from two different languages can be stored. We also made a front end GUI which could be represented in different languages.

2.0 DELIVERABLE 1

The very first deliverable for this project was to test out localization using “gettext” with an input. This deliverable helped me to understand the localization process. The testing

was performed by choosing a set of inputs against which the localization was verified. The following English poem was used as an input:

"I stood in the wind
and tried to capture it.
But when I saw the butterfly
flying flat and low-
I let the wind go,
I let it go."

The input file is shown below:

```
<?php
// I18N support information here
$language = 'bn';
putenv("LANG=$language");
putenv("LC_ALL=$language");
setlocale(LC_ALL,"bn");
$domain = 'bn';
bindtextdomain($domain, "./locale/");
textdomain($domain);
//the whole english poem goes here
echo _("I stood in the wind ");
echo _("and tried to capture it. ");
echo _("But when I saw the butterfly ");
echo _("flying flat and low-");
echo _("I let the wind go,");
echo _("I let it go.");
?>
```

Figure: PHP code for test out localization with gettext

To localize the string I made a ".po" file to translate for the given input strings. A sample code snippet is used to translate the following strings:

```
#: hello.php:12
msgid "I stood in the wind "
msgstr "Ami chilam dariye batashe"
msgid "and tried to capture it. "
msgstr "Ebog chachchilam dhorte taake"
msgid "But when I saw the butterfly "
msgstr "Kintu dekhlam jokhon projapotike"
msgid "flying flat and low-"
msgstr "Nichu diye jete"
msgid "I let the wind go,"
msgstr "Ami batash ke jete dilam"
msgid "I let it go."
msgstr "Dilam jete"
```

Figure: ".po" input file to test out localization

These translated strings looked something like the following in the poedit editor.

The ".po" file looks like this:

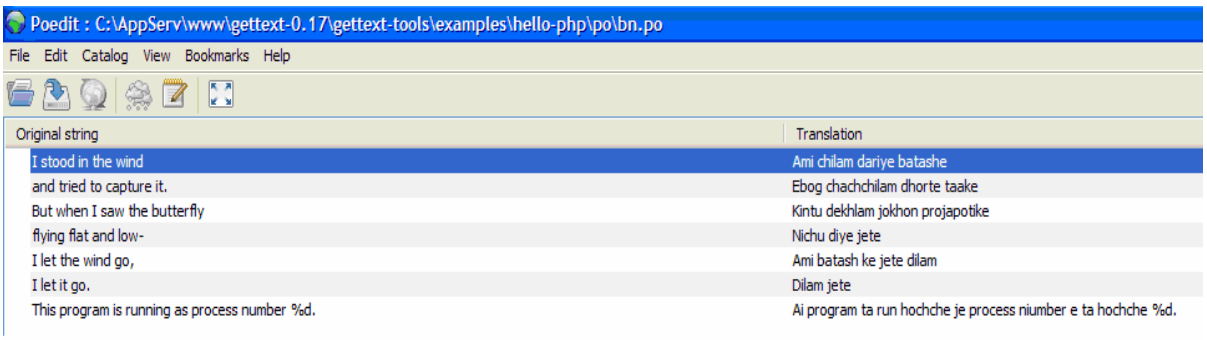

Figure: ".po" input file in poedit editor

The ".po" files were stored in the "locale" directory. Then I generated ".mo" files from the ".po" files by performing "msgfmt" operation on them. After creating the ".mo" file the browser was able to read the code and the output looked like the following.

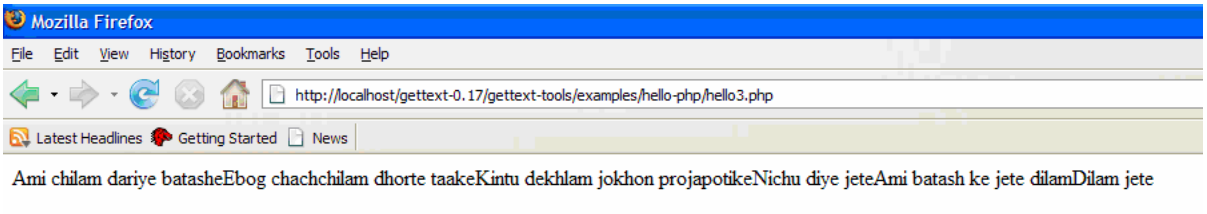

Figure: result of testing out localization with “gettext”

3.0 DELIVERABLE 2

The deliverable2 was about doing some minor modification on “gettext” code. I took the original “gettext” php code and modified the strings (piglatinized), then transformed the piglatized strings to a selected locale.

The php code for pig latinization looks like the following:

```
<?php
function __($str)
{
            $tbl = _($str);
         $words =  explode(" ", $tbl);
          // Loop through each word for translation
         for ($i = 0; $i < count($words); $i++) {
                 $temp1 = (substr($words[$i],0,1));
                 $temp2 = (substr($words[$i], 1));
                 echo $temp2.$temp1._("ay")." ";

          }

}
?>
```

Figure: PHP code for pig latinization

I have used the same old poem.php as my test file.

The test file looks like this:

```
<?php
$language = 'bn';
putenv("LANG=$language");
putenv("LC_ALL=$language");
setlocale(LC_ALL,"bn");
$domain = 'bn';
bindtextdomain($domain, "./locale/");
textdomain($domain);
include "piglatin.php";

echo __("I stood in the wind");
echo __("and tried to capture it. ");
echo __("But when I saw the butterfly ");
echo __("flying flat and low-");
echo __("I let the wind go,");
echo __("I let it go.");
?>
```

Figure: Input test file for testing pig latinization

After performing pig latinization and localization in the browser window the output looks like the following:

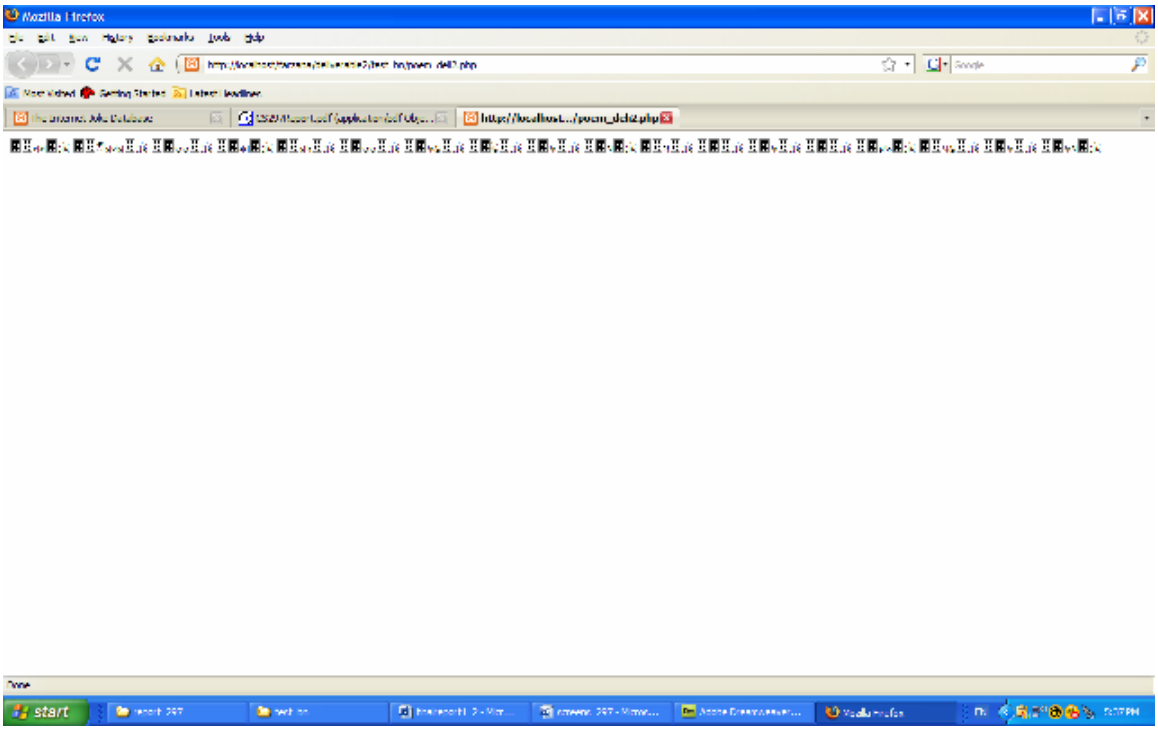

Figure: Output for the modified strings

4.0 DELIVERABLE 3

Deliverable3 was about performing some simple experiments to use the dynamic localization scheme based on database tables. Here I came out with different GUIs for different languages. The GUI can be localized to other languages by a button click. The input strings for the front end GUI comes from a mysql database table.

The set up of a database table for some specification looks like the following:

```
+------------------+------------------+------+-----+---------+-------+
| Field            | Type             | Null | Key | Default | Extra |
+------------------+------------------+------+-----+---------+-------+
| id               | int(11) unsigned |      | PRI | 0       |       |
| locale           | varchar(10)      |      | PRI |         |       |
| localized_string | text             |  yes |     | NULL    |       |
+------------------+------------------+------+-----+---------+-------+
```

Figure: database table setup specification

For two different languages (English and Bengali) the representation of GUIs looks something like the following:

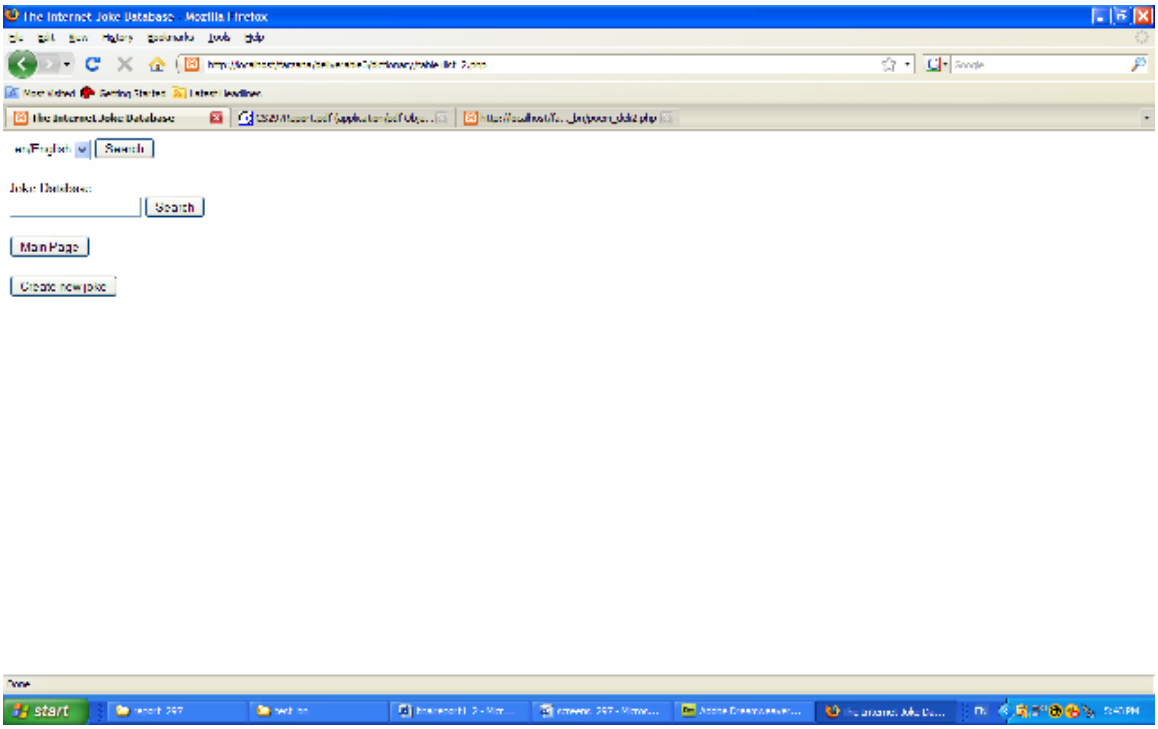

Figure: User Interface for an GUI in en/English

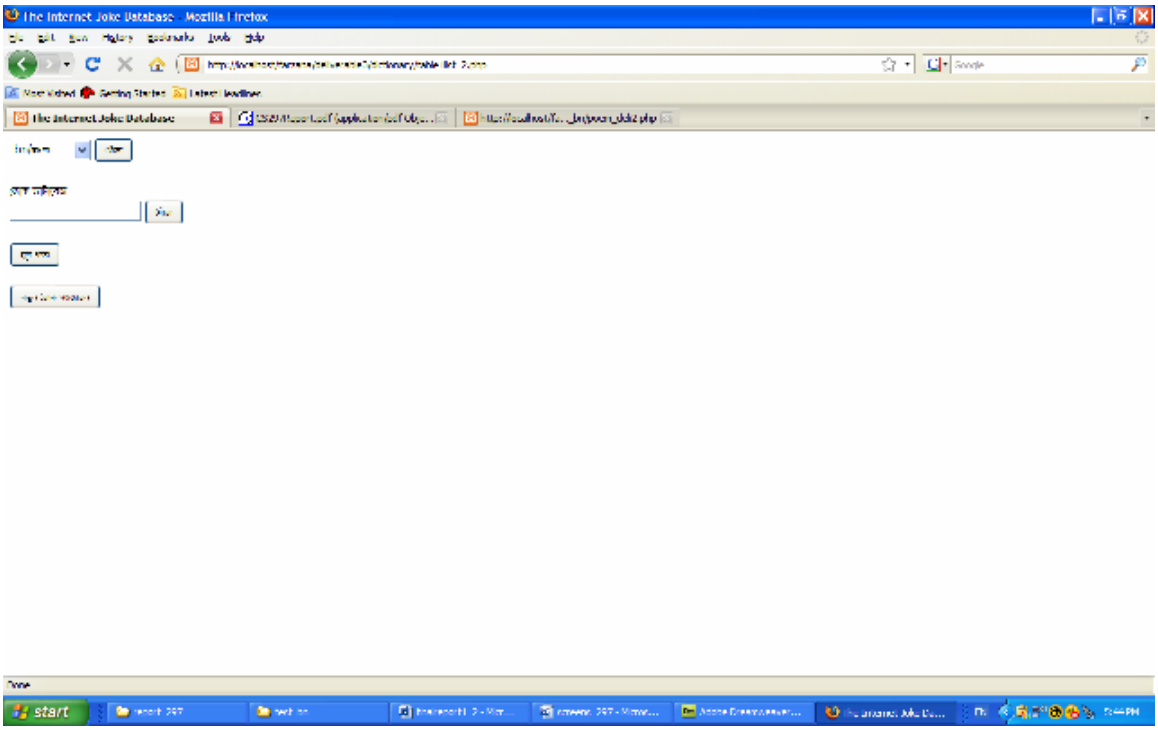

Figure: User Interface for an GUI in bn/বাংলা

The followings can be achieved from deliverable 3:

- GUI changes according to the specified locale.
- There is "joke database" set up and jokes can be extracted by interacting with the GUI
- New jokes can be added to the joke database in different locales

5.0 CONCLUSION

The purpose of this project was to find out better and more efficient way for open source localization. In this semester the deliverables helped me to get the pieces ready for the next semester's final work. These deliverables also helped me to do a very intense research on the topic.

By completing the above deliverables I was able to know more about the localization. I did an extensive research on this new and exciting topic which I believe will help me to move further with my future work.

REFERENCES

[2000] Practical Guide to Localization (Language International World Directory). Bert Esselink. Iohn Benjamin's Publishing Co. 2000.

[2000] Localization A Global Manifesto. Hines, Colin. Stylus Pub Llc. March 2000.

[2004] Technical Reports & Notes."http://www.w3.org/International/publications"

[2007] Internationalization Activity. "http://www.w3.org/blog/International?cat=33"