MULTILINGUAL USER INTERFACE FOR WEBSITE USING RESOURCE FILES

Neha Chhatwani[1], Teni Gada[2], Vani Ganji[3], Jaitheradevi Pathirapandi[4], Nishi Tikku[5]

[1, 2, 3,4]Department: MCA, [5]Professor Department: MCA, VESIT Mumbai, India
chhatwanineha@gmail.com, tenigada@gmail.com, vani.reddy007@gmail.com
jaitheradevi@gmail.com, 1102nishi@gmail.com

Abstract

Due to the rapid growth of the internet usage, new kinds of problems are emerging endlessly and one among them is language barrier among different Internet user, so it is important to build a system which will overcome this barrier. Some websites do provides language switching options but the solutions are not satisfactory as multiple pages need to be developed for multiple languages which involve lot of resource wastage.

This paper discusses multilingual implementation of website using the concept of Resource file in Asp.Net. ASP.NET and the .NET Framework ship with support for multilingual applications, namely in the form of Resource Files. Multilingual User Interface (MUI) enables the localization of user interfaces for globalized applications. MUI also supports the creation of resources for any number of user interface languages. MUI ship single core functionality to all platforms independent of UI language, which significantly reduces development and testing efforts. The most visible benefit of MUI is that multiple users can share the same webpage and view the user interface in different languages. Multilingual accessibility to Website content significantly facilitates multilingual users to meet their needs.

Keywords *- Language Barriers, Multilingual Website, Resource File.*

----------***----------

1. INTRODUCTION

"The limits of our language" means the limits of our world.- Wittgenstein, 1922, p. 149. The language barriers as natural outcomes of human communications across national boundaries not only refer to the historical dominance of the English language, but also refer to the presently increasing multilingualism on the Internet (Crystal, 2001b; He, 2008a). Increased globalization is forcing a growing number of users to interact across linguistic boundaries (Lauring, 2008). Since language affects almost all aspects of everyday life, there needs more of a focus on communication barriers by researchers (Henderson, 2005). The fact that most of the website in India are in English greatly affect majority of population of the country speaking different language. Visitor of the website may find it difficult to get necessary information regarding services provided as language barriers is one of the significant problems that users face while searching, accessing, and retrieving multilingual contents on the Internet. 75% of the world's population doesn't speak English, the websites that are only available in English have contributed to an information accessibility gap between English and non-English speaking users of Website.

Fig.1 Solution to Many Languages

2. LITERATURE REVIEW

A. Language Barriers and Multilingual Websites

Fig.2 Language Barrier Issues

Language barriers would create obstacles to effective communication on a website, such as:

- Languages around the world differ in display, alphabets, grammar, and syntactical rules. For Exampleample, such languages as European French and German do not have a one-to-one mapping characters, while most non-Latin character-based languages, such as Hindi, Marathi etc. do not even use the concept of lower- and uppercase (MSDN Museum, 2007).
- Many idioms and Examplepressions may have different meanings when literally translated into another language. There also Exampleist regional variants of the same language.

B. Multilingual Museum Websites

A multilingual Museum website provides the same information in different languages. For the online services provided by Museums via their websites, multilingual accessibility is significantly important in meeting the needs of diverse Museum clients by making information available in as many languages as possible and to overcome language barriers. Therefore, multilingual websites play a strategic role in the quality and effectiveness of the information and services provided by Museums on the Internet.

Fig. 3 Multiple Languages

A survey on public museum websites in India reveals that very few of them had multilingual pages and resources for users to select. In September 2006, a study was conducted on the availability of multilingual websites of cultural establishments in Delhi. The survey found that 16 out of 18 public museum websites were monolingual; representing 93% of the total number of such websites; and only 2, or 7%, of public museum websites are multilingual.

From June 2004 to May 2005, MINERVA Plus (2006) conducted a major survey on the multilingualism of the websites of 265 museums, 138 libraries, 98 archives, 65 cultural websites, and 129 other websites in 24 European countries. The findings were very interesting as follows: Of the 695 websites, 179 of them were monolingual, 310 were bilingual, 129 were available in 3 languages, 26 were available in 4 languages, 14 were in 5 languages, 10 were in 6 languages, 4 were in 7 languages, 3 were in 9 languages, and 1 was in 34 languages.

From the above we could see that multilingualism of museum websites as an emerging issue has drawn attentions from researchers but still calls for further investigations.

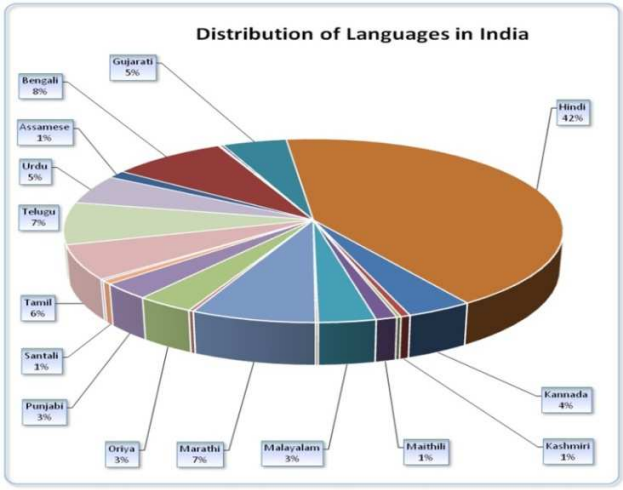

Fig.4 Statistics showing language speaking proportion

3. OUR METHOD OF IMPLEMTATION

3.1. Using Resource files in Asp.Net

Resource file is Asp.Net can be used to store images, XML files, XSL files, XSD files, VBScript files, sound clips in different languages (or even using the new .NET Speech SDK grammar files), and even video clips that are loaded dynamically. Resource files are typically used to store user interface strings that must be translated into other languages. This is because you can create a separate resource file for each language into which you want to translate a Web page. Global resource files are available to any page or component in your

Web site. Local resource files are associated with a single Web page, user control, or master page, and contain the resources for only that page.

3.1.1 Resource File Methodology:

Local resource is specific to a single Web page and used for providing versions of a Web page in different languages. Local resources must be stored in App_LocalResources sub folder. It must be named in format <WebPageName> [.language / language and culture].resx.

Example: Default.aspx.resx- Base resource file. This is the default, or fallback, resource file.

1) Generating Default Resource File using VS2008:

Default Resource files are generated using following steps
Step1: In Visual Studio designer, click the designer surface or a control.
Step2: Select Tools --> Generate Local Resource.
Step3: An XML - based local resource file for the Web page in the App_LocalResources folder with TExamplet and ToolTip values for all Exampleisting controls on the page, as well as the page title will be generated.

2) Generating Resource Files for Other Cultures:

In Solution Exampleplorer following steps need to be followed:
Step1: Right click the 'Default.aspx.resx' file and click the Copy
Step2: Right click the App_LocalResources folder and click the Paste
Step3: Right click the 'Copy of Default.aspx.resx' and click the Rename
Step4: Type the new name for the resource file that includes the new language and culture code before the Exampletension.
Example: 'Default.aspx.fr.resx' to create a French language version of 'Default.aspx' page.
Step5.Double click the resource file to open it in Visual Studio. Update the values of each resource for the new culture, and save the file

3) Testing Resource Files for Other Culture:

ASP.NET automatically determines the user's preferred culture based on information provided by the Web Browser. To test other culture we need to update the preferred language in Web Browser as
Step1: In Internet Exampleplorer, select Tools --> Internet Options
Step2: Under General tab, select Languages
Step3: In Language Preference dialog box click on Add button, Select the language from Add Language dialog box and click on Ok button.
Step4: In Language Preference dialog box, under Language list make newly added language on top using Move Up , and then click on Ok.
Step5: Now Webpage will be displayed using selected language resource.
Step6: In Language Preference dialog box, under Language list make newly added language on top using Move Up , and then click on Ok.
Step7: Now Webpage will be displayed using selected language resource.

4) Using Resource to Set Control's Property Values

By Implicit Localization:

If local resource files are created in Application, then implicit localization can be used to fill the property values for the control. To use implicit localization, following naming convention must be used for resources in local resource file key. Property where key: Any name for the resource Property: Property of the control that we are localizingExample: If we are creating resource for control label named lblErrorMsg, you need to create following key/value pairs in local resource file

lblErrorMsg.TExamplet="ErrorMessage"
lblErrorMsg.ForeColor = "Red"
Step1: In aspx page, we need to use a special meta attribute in the markup for the control to specify implicit localization as-
<asp: Label ID="lblErrorMsg" runat ="server" meta: resourcekey =" lblErrorMsg" TExamplet ="Label"></asp: Label>
Step2: .No need to Exampleplicitly specify which properties are localized.
Step3: The resourcekey value matches a key in resource file.
Step4: At run time, ASP.NET matches resources to control properties using the resourcekey, if property value is defined in resource file, ASP.NET substitute the resource value for the property.

3.1.2) Retrieving Resource Values Programmatically:

Call the GetLocalResourceObject () or GetGlobalResourceObject () method to read specific resource from local or global resource file, respectively.

```
Example: string localresourcestring = string.Empty;
    // Get the local resource string.
    try
    {
   Localresourcestring =(String) GetLocalResourceObject
("LocalResourceString1");
    }
    catch
    {
            Localresourcestring = "Could not find local
resource.";
    }
    // Get the global resource string.
```

```
try
{
 globalresourcestring =(String)
GetGlobalResourceObject ("MyResource",
"GlobalResourceString1");
}
Catch
{
   globalresourcestring = "Could not find global resource";
 }
```

3.1.3 Editing Resource Files after publishing:

The beauty of the resource file in ASP.NET is modifying the resource file after the application is deployed on a server without re-compiling the entire application itself. Resource file only under the App_LocalResources will get published as a raw .resx files on the server as these are not compiled. These resources files can be edited on the server as they are compiled during runtime. Files under the App_GlobalResources folders are compiled into individual resource specific dll's and published on the server, so we cannot edit resource under this folder

B. Using Sessions

HTTP is a stateless protocol. This means that a Web server treats each HTTP request for a page as an independent request. The server retains no knowledge of variable values that were used during previous requests. ASP.NET session state enables us to store and retrieve values for a user as the user navigates ASP.NET pages in a Web application. ASP.NET session state identifies requests from the same browser during a limited time window as a session, and provides a way to persist variable values for the duration of that session. By default, ASP.NET session state is enabled for all ASP.NET applications. We use session variable Example tensively to store the value of language selected by the user so as to retain the language preference of the user when he/she navigates through different pages of the website.

Retaining language preference is made possible by using the concept of session is ASP.NET.

```
Dim val As String = Session ("uic")
UICulture = val
Culture = val
Thread.CurrentThread.CurrentCulture = _
CultureInfo.CreateSpecificCulture (val)
Thread.CurrentThread.CurrentUICulture = New _CultureInfo
(val)
MyBase.InitializeCulture ()
```

CONCLUSIONS

In this paper, we addressed the topic of language barriers with a reference to the multilingualism of the Internet and the Web. As an initial attempt, we motivate and solve the problem of supporting efficient access to web content for achieving effective utilization of web contents by visitors of different language. We first give a basic insight about the current trend of websites and show that by following the same Example listing shortcoming, barriers and its influence on the multilingual web users. We discussed the need for a multilingual website by the diverse population of each city and how the multilingual websites meet such a need. We discussed the challenges brought up by language barriers in terms of providing web contents and services on the websites, and the opportunities to bring solutions for overcoming language barriers on websites. Finally, we briefly summarize the significance of this study.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers of our website who have provided insightful and constructive comments on and suggestions for this paper.

REFERENCES

[1] http://msdn.microsoft.com/en-us/library/ms178581 (v=vs.100).aspx
[2]http://forums.asp.net/t/1540920.aspx
[3]http://www.programmingforums.org/post182345.html
[4] Matthew MacDonald, "Beginning ASP.NET 3.5 in VB 2008", 2nd ed.
[5] Robert Hopkins, Jr., "Benefits you can count on, headaches you can avoid"