

# Interacting with Dynamic Web Portals in Local Language

Ch.JyosthnaDevi ,Santosh Kumar shah ,J.Devi Durga, K.Ruth Ramya,N.Vahini,Gyan pandy

**Abstract-** Today and Tomorrow , the most advanced technology is Information Technology (IT) . But, the applications of IT are being limited to only those who are familiar in English. It is not useful for those who are unknown in digital literacy. In an increasingly interconnected world, the interactions among devices, systems, and people are growing rapidly. At present the repositories in Internet are mainly in English, as a consequence users unfamiliar in English are not able to get benefits from Internet. Although many enterprises like Google have addressed this problem by providing translation engines but they have their own limitations. One major limitation is that translation engines fail to translate the dynamic content of the web pages which are written in English in web server database. We address the problem in this work and propose a user friendly interface mechanism through which a user can interact to any web services in Internet. We illustrate the access of Andhra Pradesh State Road Transport Corporation System and interaction with Wikipedia English Website signifying the efficacy of the proposed mechanism as two case studies.

**Index Terms** - Human computer interaction,Information technology, Internet, Information retrieval, Ubiquitous computing, Virtual Keyboard.



## 1. INTRODUCTION

India is a highly multilingual country with eighteen constitutionally recognized languages and several hundred dialects & other living languages. Even though, English is understood by less than 3% of Indian population, it continues to be the de-facto link language for administration, education and business. Hindi, which is official language of the country, is used by more than 800 million people. According to UNESCO report [1], about 64% of population of the globe is English illiterate. The percentage is more in developing countries like India, China, and Pakistan etc. where nearly 76% people are English illiterate. It is obvious that these people are deprived from the advantages of Internet because majority of web pages (nearly 45%) are in English ([2], [3]). Hence, there is a need to bridge the digital divide exists since beginning of IT revolution i.e. the last decade of the previous century. Let's discuss about challenges ahead to access the Internet repository.

In present scenario, users who are familiar with users languages and less familiar with English face difficulties in accessing the web services.

- 
- Santosh kumar shah is currently pursuing Bachelors degree program in computer science and Engineering in KLCE , India, E-mail:shah.santoshcse@gmail.com
  - J.Devi Durga lakshmi is currently pursuing Bachelors degree program in computer science and Engineering KLCE,India, Email: devi.jangala@gmail.com
  - Ch.Jyosthna Devi, Asst Prof, Dept Of CSE, K L University

The traditional web services (like Wikipedia search engine and many other) generate dynamic web page [4] with respect to query given by the user in English language as they maintain their database in the same language through remote web server.

Thus, the services are lacking of support to user's query in Indian languages and as a result, unable to produce dynamic web pages in any language excluding English. It may be also observe that many translation engines like google translate([5]–[11]) which convert web pages from English to Indian languages, have addressed the problem for static content of a web page.

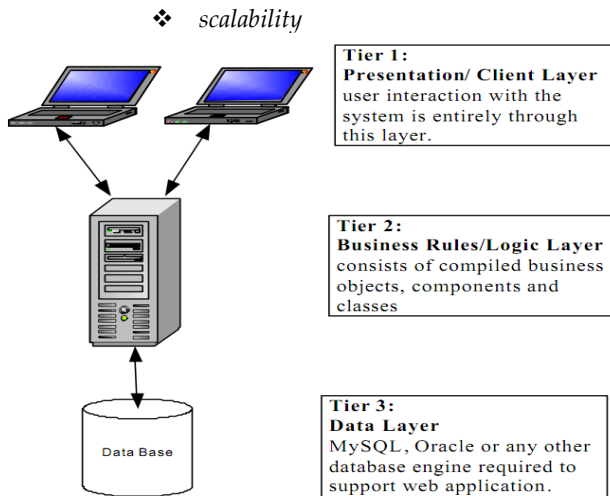
For dynamic web page content, the success rate is very poor. This specific challenge has been addressed in this work. We propose [13] a mechanism, called "Two-handed Interaction" which enables a user to interact with the dynamic web pages in user's required language only and the results returned during interactions are displayed in same language.

## 2. Standard architecture for web-based applications

This system is designed based on a traditional three-tier architecture used by many web applications. Three-tier architecture includes a presentation layer, business rules/logic layer, and the data layer. The three-tier architecture is shown in Figure 1.

The three-tier architecture is generally used when an effective distributed client/server design is needed that provides

- ❖ increased performance
- ❖ flexibility
- ❖ maintainability
- ❖ reusability and



**Figure:1 Standard architecture for web-based applications**

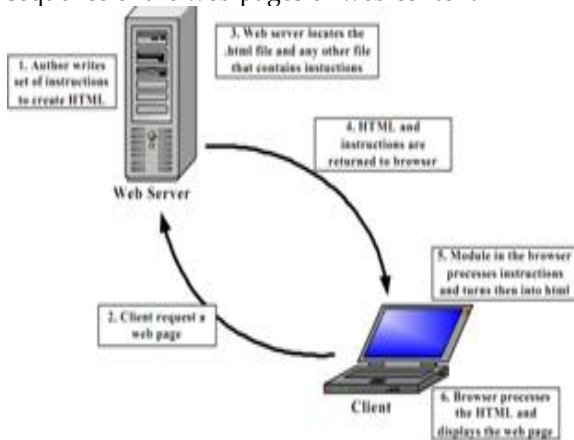
### 3. Implementation Technologies

By typing a URL (Uniform Resource Locator) into the address box of the browser the communication between a browser and a web server is started. Each conversation consists of two pieces:

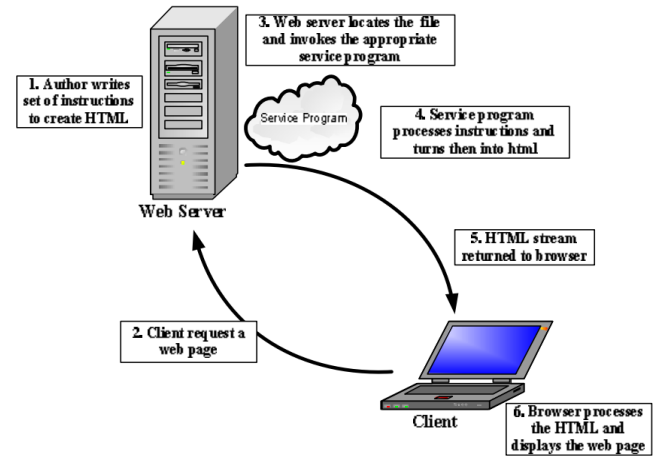
#### Dynamic web page

In a dynamic web page content (text, images, fields, etc.) on the web page can change, in response to different contexts or conditions. There are two ways to create this kind of web pages:

1. Using client-side scripting to change interface behaviors within a specific web page
2. Using server-side scripting to change the sequence of the web pages or web content



**Figure:2 Client-Side Dynamic web page**



**Figure:3 Server-Side Dynamic web page**

### 4. Google Translation

Google Translate is a free, web-based and statistically-based machine translation service provided by Google. It enables to translate section of text, document or webpage, from one language to another.

#### 4.1 Google Translate API

The Google Translate API lets websites and programs integrate with Google Translate programmatically. During the project development phase Google provides two versions of API. The version 2 is the latest available version of the Google Translate API. We decided to use version 2 within the project. After that point whenever we refer to translate API in this report, we mention version 2. One needs a Google account to use the translation service. Because, the Translate API requires the use of an API key and it can only be received from Google APIs console. There are two ways to invoke the API: Using REST directly or using REST from JavaScript (This does not require server-side coding.). JSON is used by Google as a data format.

#### 4.2 Limitations

When one uses the Google API, it is required to accept the terms and conditions. Here are the most important limits for our project:

1. Every request text to be translated can be up to maximum of 5000 characters long.
2. Daily limit is 100.000 characters per API key.
3. Continuous translation requests successively results in Suspected Terms of Service Abuse.
4. Batch requests are against Terms of Service.

## 5. Tricia

Tricia Platform is an Open Source platform for developing dynamic web applications. It is built in Java and realizes the model-view-controller pattern, which consists of the following components:

1. An abstraction for defining control flow.
2. A templating language
3. An object/relational persistence mapping

It has to be emphasized that what has been described so far is called Tricia Platform, since it is a generic platform for building arbitrary dynamic web applications. Tricia Platform is implemented in the Eclipse project Toro. In addition to the pure infrastructure, the toro project already comes with the basic asset types Person and Group, which are required by almost any dynamic web application. Additionally, Toro comes with an existing basic layout. On the other hand, there are concrete plugins built on top of the Tricia platform and the basic Tricia plugin. Combining these plugins results in the Tricia Application.

A typical configuration of Tricia consists of the plugins Toro, File, Wiki, and Blog. The distinction between the platform and the application is similar to Ruby on Rails as a platform and applications like Basecamp or Backpack, which are built using Ruby on Rails.

### 5.1 Existing System

The present internationalization system in Tricia is not fully tested and does not perform its full intended functionality. It is a partially implemented system. The internalization features present in Tricia are:

#### 1. Language Handler:

The Language Handler handles the change of language parameter as triggered by the user action. The Language Handler identifies the user chosen language from the url parameter and sets in to the session variable. The session variable is set when the user clicks his/her preferred language on the Tricia page. The languages are identified with the flags as icons.

#### 2. Translator Configurator:

The Language flags on the Tricia pages can be configured through the Translator configurator tool of Tricia Platform. This tool provides away to add the desired languages (according to the multiple languages

required to be supported by the Tricia). A language can be added through the configurator tool.

The Internationalization plugin for Tricia has to

1. Identify the texts appropriate for translation.
2. Store and handle the transalted text.
3. Update the Tricia Web page according to the language chosen by the user.
4. Regularly update the translations.

### 5.2 Tricia Technical Design:

After we discovered what to be translated, we designed our translation system. It is depicted in Figure 6. We try to design the system component based. As we mentioned before, static texts are not stored in database in the Tricia. However, we decided to save translated texts into database because of three reasons: Performance issues, Google Translate API limitations and memory consumption. If we think translation as a kind of computation, we can say that we use optimization technique so-called memoiza-tion<sup>18</sup>. We translate the texts once and store them. Whenever we need a translation of a text, we look up from database table. On the other hand, this method reduces API calls and helps us not to exceed API limits. The last but not least is memory consumption. The static texts were stored in the files and when Tricia started to run, the texts were copied into memory in order to benefit of speed of memory operations. This was acceptable for one language but what happens when Tricia supports many languages? This increases memory consumption factor of supported number of languages. Scalability of Tricia was undermined due to the internalization. Storing the static texts into database enables us to overcome all these problems.

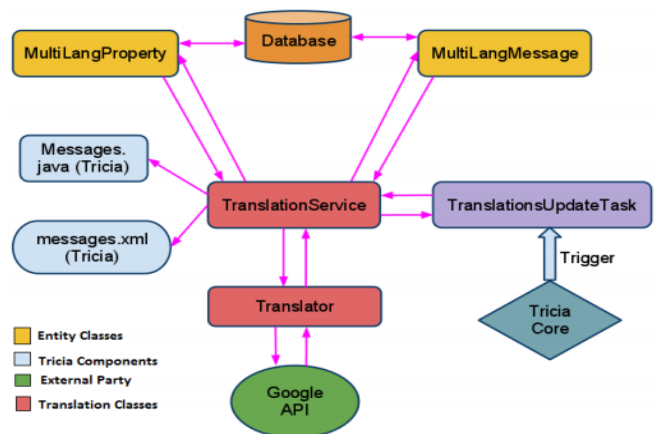


Figure: 4 overview of Translation system

## 6. System Overview

In this section, we discuss our proposed approach to interact with a web service in user's language. Let user's language be L. **A framework of our approach is shown in Fig.5.** The framework consists of two major components: RTR and IHDD, which are discussed in the subsections below. The RTR (Retrieve, Translation and Render) module searches the Internet for the web page with respect to request given by the user. After getting the requested page, it translates the retrieved web page into language L and renders the resulting web page in proper manner to the client machine. This module consists of three sub modules namely Retrieve, Translation and Render. The Retrieve sub module retrieves a web page as requested by the user and separates the content of web page (by using HTML parser) into HTML tags and English text. Proper indexing of links in the page is done and maintained using a table called Index Table. The Index Table handles the layout and links present in the original web page. The Translate sub module takes the extracted English text from the Retrieve sub module and converts the text to the language L. The Render sub module fully furnishes the web page in language L. Once text is converted, it merges the content in language L and set HTML tags using Index Table. In fact, the Render sub module recreates the web page in user's language having same look as the original web page.

The IHDD (Input Handler and Data Dispatcher) module is responsible for converting users input from language L to English and forward them to original web server. The module has been subdivided into two sub modules namely Input Handler and Data Dispatcher. The Input Handler first extracts the input data from web page in language L and then converts the data from that language to English. The module then invokes the Data Dispatcher which performs the authentication needed to access the web site and finally regenerates the query to be posted to original web server in English. The working of

the module is described as follows. User enters the URL of a web page what he wants to get serviced. The proposed interface then invokes the Retrieve sub module within RTR, (Step 1). After that, Retrieve module searches the specified web page in the Internet and result is returned back to the same module (Step 2 & 3). Once the web page is fetched, the Retrieve module separates the HTML tags and English text from the web page. The separated English content is further sent to Translation module (Step 4) for converting to language L and stores the link information in Index Table for proper maintenance of the web page (Step 6). After the completion of text conversion, Render module takes the input from Index Table and Translation module to merge the content of English text and HTML tag and generates the virtual web page in user's language (Steps 5, 6 and 7). This completes the first phase of converting the requested web page in language L.

In second phase, user gives input in his language by filling forms which appears on the virtual web page in L with the help of virtual keyboard(Step 8). Input Handler then extracts the text entered by user (Step 9) and calls Translation module to convert the text from language L to English. After that, this generated English text and the virtual web pages are sent to Data Dispatcher module (Steps 10 and 11) which regenerates the query in English, handles all the authentication needed to post the web page in Internet and then invokes the Retrieve module (Step 12). Now the task of Retrieve module is to process the request and invoke the remote server which accesses the database (database contents are in English). The result returned from the remote server has finally been stored within the Retrieve module (Step 13 and 14). The Retrieve module then sends the result to the Translation (Step 15) and Render (Step 16 and 17) modules which generate web page in language L. This gives illusion to user that the result which is displayed on the virtual web page fetched exclusively in user's language instead of English.

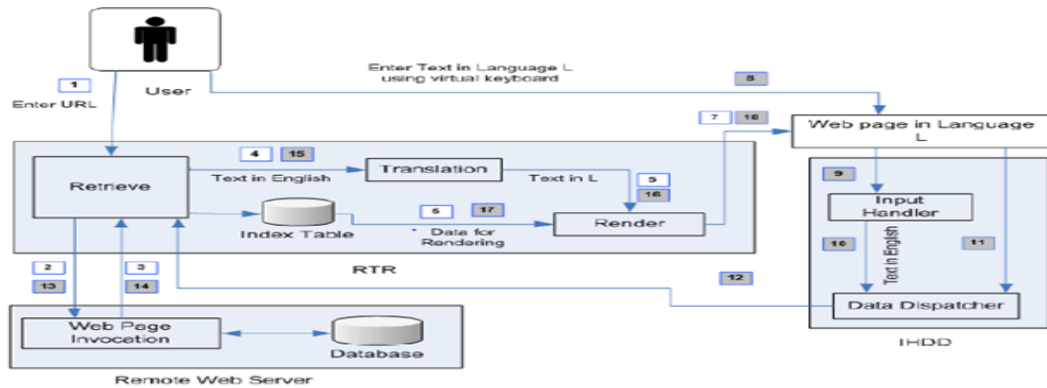


Figure:5 Framework of two-way interaction

## 7. CASE STUDY

In this section, we illustrate our proposed mechanism with two case studies.

### 7.1 Andhra Pradesh State Road Transport Corporation System

We consider one of the popular web services in India, "Andhra Pradesh State Road Transport Corporation System (APSRTC)". Our proposed mechanism provides an interface between the user and APSRTC in Internet maintained by Andhra Transport. The interface gives a flavor to the user that the entire APSRTC web page is written in his language. It also provides facility to the user for giving input and getting output in his language in his language. In this study, we explain the accessing the APSRTC in Hindi language (National language of India). It is not necessarily limited to Hindi. In fact it can be applied to any language, of course with their

corresponding translation scheme in needed.



Fig:1 Accessing Dynamic web page of Andhra transport in English language

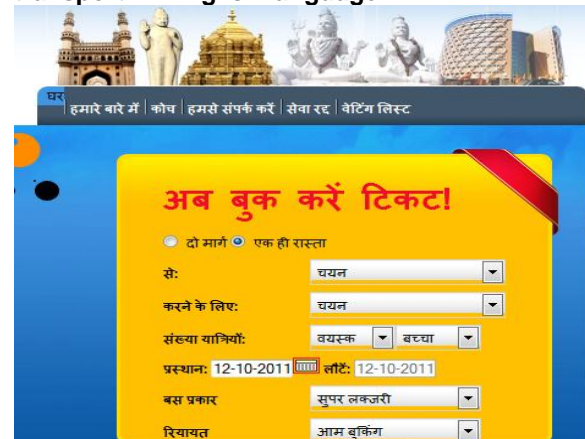


Fig:2 Accessing Dynamic web page of Andhra transport in Hindi language

### 7.2. Wikipedia English website

Searching for the word कृषि in English Wikipedia "http://en.wikipedia.org/wiki/" using Google website translation service, will return only those page written in Unicode. As the word कृषि doesn't exist in English Wikipedia it will not return any result but when the word "Agriculture" is given it will display the English pages containing the word "Agriculture".



Fig:3 Wikipedia Search Engine

Fig:4 Result when the user gives input In English Word



Fig:5 Result when user gives input in Hindi under Google translation for word कृषि.

### 4. DISCUSSION

In this work, we have addressed the problem facing by English illiterate people to access Internet where majority of the pages are stored in English. Our work is a contemporary and first of its kind. The work also addresses many limitations, which Google is yet to address. Google has web page translation scheme called Google Translate . This translation scheme is partial and more importantly one way. As an example, to translate APSRTC web site using Google's translation, user must give the station name in English. If user gives the input in English, site searches the BUS between the given stations and resulting web page is rendered to user in English only (as Google loses the control to translate). When user gives the source and destination name in Hindi Unicode like source name and destination name, Google gives unauthorized invocation error and unable to translate the page, neither in English nor in hindi. We have tested our mechanism with more than 35 popular web sites with two way interactions and results are error free and satisfactory. We have tested our approach with Hindi and Bengali and it is applicable to any language provided that accurate translation scheme is available in that language.

Interacting Dynamic Web Portal in local Language plays a great role, as it will minimize the gap between the Internet and user due to language barrier . It helps the users to access Internet in their day to day life without worrying much about the language web page is originally written. This service provides the Internet content to millions of people who might not have good capability to read the web content in English.

### REFERENCES

- [ 1 ] "Unesco, international literacy statistics a review of concepts, methodology and current data," <http://unesdoc.unesco.org/images/0016/001628/162808e.pdf> .
- [ 2 ] "India world's second largest english speaking country," <http://tesolindia>.

c.in/EnglishTeachingIndustry/india-worlds-second-largest-english-speaking-country

[3] "Languages and cultures on the internet study 2007," <http://dtil.unilat.org/LI/2007/ro/resultados-ro.html>.

[4] "Wikipedia, dynamic web page," [http://en.wikipedia.org/wiki/Dynamic\\_web\\_page](http://en.wikipedia.org/wiki/Dynamic_web_page)

[5] "Googletranslate," <http://translate.google.com>

[6] "Translate a block of text," <http://in.babelfish.yahoo.com>

[7] "World lingo, free website translator," <http://www.worldlingo.com/en/websites/url-translator.html>

[8] "Freetexttranslation," <http://www.freetranslation.com>

[9] "Free automatic translators, machine translations comparison tests,"

<http://www.humanitas-international.org/newstran/more-trans.htm>

[ 10 ] "Language translation, translate phrase or word," <http://www.translation.langenberg.com>

[ 11 ] "Websites translator, translate your website from english," <http://www.websitestranslator.com> (2010).

[ 12 ] "Taming the beast, web page language translation," <http://www.tamingthebeast.net/article-s6/page-language-translation.html>

[ 13 ] "Ability, website translation and localization," <http://www.localization-translation.com/translation-ocalizationservices/web-sites-localization.html>

[ 14 ] "Accessing Dynamic web pages in user Language" [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5783859](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5783859)