Implementing a Multilingual Capability to a Multimodal Platform: Prime III an Electronic Voting System

Wendy C. Velasquez Ebanks
Undergraduate Student
University of Maryland University College
wvelasquezebanks@student.umuc.edu

Juan E. Gilbert
Adviser, Endowed Professor and Chair
University of Florida
juan@ufl.edu

Elizabeth Mathews
Phd Student
University of Florida
lizmatthews@ufl.edu

Wanda Eugene
Post Doc
University of Florida
weugene@ufl.edu

University of Florida Gainesville, FL 32611
University of Maryland University College, University of Florida

Abstract

Prime III is a secure accessible, multi-modal electronic voting system that emerged from an initiative to create an innovative voting system that could be accessible to everyone. Using touch screens, voice, typing, and switch devices, Prime III allows a wide range of citizens to take part in the voting process. This study was conducted with the purpose of improving the current experience of Prime III. Although it is a multi-modal platform, it lacked a Multi-Language ability. Adding this feature on the system will offer future users the ability to vote and understand important statements of the process in their native language at the moment to practice their right to vote. Without this feature, voters with limited English proficiency may disenfranchise, unable to understand complex statements of importance while casting their vote. Now, with the test and development of a language object that will be incorporated into the Prime III, will allow more people to vote and feel confident when casting their vote. Therefore, this feature is going to have a great impact on the users of Prime III.

I. Introduction

Prime III implements a multimodal design: "a product and environment to be as usable as possible by as many people as possible regardless of age, ability or situation". [1] However, it does not have a multilingual capability. A multilingual system is the best way to cross language barriers, and to capture the attention of those that get discourage because they are not proficient in another language [2].
The main focus of this research project was to analyze, document, design, and implement a strategy that could allow the incorporation of another language besides English into the Prime III system. For the purposes of the study Spanish was the language selected for the demonstration. Given that Spanish is the most commonly spoken language in the US, after English, by having 37 million speakers as of September of 2013 and increasing [3], [6]. Having an electronic secure voting system that provides a multilingual accessibility along with its multimodal access will have a remarkable impact in a society and working towards the possibility of increasing the capability to any other language in the future (See Figure 1).

II. Related Work

Researchers have studied different types of methodologies to implement translation in multimodal platforms facing different types

of challenges such as the evaluation and analysis of data structures to find out which one would best fit to the complexity of the system they are working on in order to integrate the different software components in addition to the electronic and mechanical devices that system interacts with [8], [9]
There are diverse aspects to keep in mind, when deciding for one approach or the other when referring to translation methods and multimodal interaction has been studied for several years; (Oviatt, 1999 and Oviatt et al, 2000). Most papers on user studies report experiments that were carried out with Wizard-of-Oz systems and professional users who manipulated objects on large terminal screens (Kehler et al., 1998, Martin et al., 1998, and Wahlster et al., 2001) [4].
Other techniques of translation on Multimodal platforms include: Tokenizer dequeue on which all tokens are specified off-line in a form of regular expressions. Then a Finite State Machine (FSM) compiler is used for the construction of a tokenizer, which is the most similar to what was implemented for the translator for prime III [7].

III. Methodology

Study of Prime III structure:
The Prime III is coded mainly in JavaScript; it uses an HTML, CSS, and PHP programming languages. In order to understand its complexity, and to find ways to develop a strategy for translation there were two main starting points:

- Test Prime III system as user.
- Learn back end of Prime III to design and create concept map that shows the relationship of each file contained on it.

The complexity of its structure presented several challenges, and in order to understand the inner functionality of the system; some of its coded functions had to be tested and documented separately. *Duplicating the GUI code of Prime III translated:*
This method consisted in translating label by label of the Graphical User Interface to the actual meaning in Spanish. However, that was a very ineffective method because it will produce a duplication of the entire interface code of Prime III, which is already long enough (See Figure 2).

So, research of tools such as a piece of code or APIs that could be used to create the segment that will perform the interface translation on Prime III. The study of the structure of Prime III is coded with the four programming languages previously mentioned. the first step with coding was to find libraries and functions that would work for Prime III. Before testing any of these pieces or APIs the local-host had to be configured; otherwise, none of them would work properly. The JavaScript programming language does not provide any built in function for translation. Instead, it provides a library that uses online libraries such as the one called "localized.js" to translate entire websites [5].

Test API code:
Part of this study was to test its functions and certain parts of Prime III separately to verify the effectiveness of one of the strategies implemented such as using the Google API for translation, and After trying the inefficiency on translating the tags by duplicating the code this was the most effective way. The Google API took three different phases.

- Phase 1: Display the testing with the initial screen of Prime III and the modification resulting from using the Snippet code from Google (See Figure 3.
- Phase 2: Demonstrate the first attempt of working with Google voice and translation recognition API and its difference in code (See Figure 4.
- Phase 3: Shows that in order to manipulate and customize the translation methods of Google API the developer must have a paid access Key (See Figure 5).

The snippet changed the configuration of the interface of Prime III, and also it needs of the Internet connection in order for the translator to work.

The next step on using Google's API was to find how to manipulate its functions for translation and customize them in a way that could work for Prime III. However, that is not a possibility unless you buy the access key that Google provides to work with its functions, similarly, happens when using Microsoft Translation API in to manipulate an Access Key is required for it (See Figure 6).

IV. Results

The translation object was completed and will be soon implemented into the Prime III system. Although Prime III system is complex in functionality, it is not in the vocabulary used to interact with the user. A preliminary analysis evaluation was made to find other methods that could improve the translation methodology resulted from this research.

The results are shown on Figure 7

The best strategy to use for this purpose was to create a unique library and glossary of phrases in another language for Prime III. We utilized Spanish since that is the pilot second language to be added to the system. The first step was to develop a strategy on creating an object that will hold the language equivalences and will be linked with the access code entered (See Figure 8).

Once the structure was already design and plan, the following step was to create a small glossary table that contains the English phrases of Prime III interface and its Spanish translation in detail such as the table on figure 9.

All translations had to verify for correction with online dictionaries, and the HTML ASCII codification particularly for vowels with tilde when translated into Spanish. The template developed from the proof of concept will be used to include additional languages, to extend the accessibility benefits of Prime III to a broader range of voters (See Figure **??**).

Testing Functions and parts of Prime III separately to verify effectiveness of the strategy implemented:
Functions and small parts of Prime III code were taken to create a demo that could simulate the actual functionality of prime III once the translation object is integrated in the system. This was mainly done for testing purposes and to find out what the result would be once is implemented.

Learning and Testing Web Speech Recognition: Mozilla corporation recently release the Web speech Recognition Feature for its browser [11]. In this case the main purpose was to learn, test, and try some of the functions included in the Web Speech Recognition API from Mozilla. The only two parts of the API tested were only WebRTC "Still Photo" with web browsers through the Web Speech API, and the communication of one Web browser to another such as Firefox communicating with Google Chrome or Internet Explorer. through the "WebRTCDataChannel".

V. Discussion and Design Recommendations

Future work includes the actual implementation of the translation object into Prime III, now that it has been proved and tested. As a result, continuous evaluation will be an essential part of this research which proves to be a successful approach to enhance progress and to improve the system in various areas that relate to the translation feature. Being able to split the language script into different files has the advantage that any developer can concentrate its efforts on a single file, and test its performance within a complete system, or using a part with missing sections to test its functionality.

VI. Conclusion

Overall, this summarizes the challenges encountered in the development of multilingual feature for a multimodal platform. The learning experience of studying a multimodal platform such as Prime III, and the benefits that could bring to a community. Understanding user needs, the need, the targeted user, and adapting systems that are specialize to the user.

Definitely, the challenge of developing human-centered applications lies on understanding the need, the targeted user and develop systems that can adapt and specialize to the user [10], which has been a learning lesson throughout the research experience.

VII. Aknowledgements

Special thanks to the Computer Research Association for Women (CRA-W), The Distributed Research Experience for Undergraduates (DREU), the Coalition to Diversity Computing (CDC), The National Science foundation (NSF), they made possible this experience. In addition, I would like to thank my Adviser Dr. Gilbert and his team at my host institution University of Florida because without their constructive support and assistance I would not have had such an amazing and productive experience.

References

[1] Prime III: *"About Prime III"*, `http://www.primevotingsystem.org/`.

[2] Rachel Evans, Joanna E. Cuevas Ingram, "Lawyers' Committee for Civil Rights of the San Francisco Bay", *Voting Rights Barriers and Discrimination in Twenty First Century California 2000-2013* P 1-50. 2014.

[3] Mark Hugo Lopez, Ana Gonzalez-Barrera "Pew Research Center" *What is the future of Spanish in the United States?*, P. 2013.

[4] Almeida, Luis and Amdal, I and Beires, N and Boualem, M and Boves, L and den Os, E and Filoche, P and Gomes, R and Knudsen, JE and Kvale, K and others, "International CLASS Workshop On Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems" *Implementing and evaluating a multimodal and multilingual tourist guide*, P. 2002.

[5] *https://github.com/coderifous/jquery-localize*

[6] Laura B. Shrestha, Elayne J. Heisler, "Congressional Research Service" *The Changing Demographic Profile of the United States*, P 22-24. 2011.

[7] Matej Rojc, Izidor Mlakar, "InTech Speech and Language Technologies", *Multilingual and Multimodal Corpus-Based Text to Speech system -PLATTOS-* , P 129-154. 2011.

[8] Yosr Jarraya, Mourad Debbadi, "Springer", *Quantitative and Qualitative Analysis of SysML, Activity diagrasm*, P 399-419. 2014.

[9] Eytan Adar, Michael Skinner, Daniel S. Weld, *Information Arbitrage Across Multilingual Wikipedia* P 94-103. 2009.

[10] Ahmed Elgammal, "ACM", *Human-Centered Multimedia: Representations and Challenges*, P 11-17. 2006.

[11] Mozilla, "Mozilla Corporation", *Web Speech Recognition (Web RTC*, 2015.

VIII. Appendix

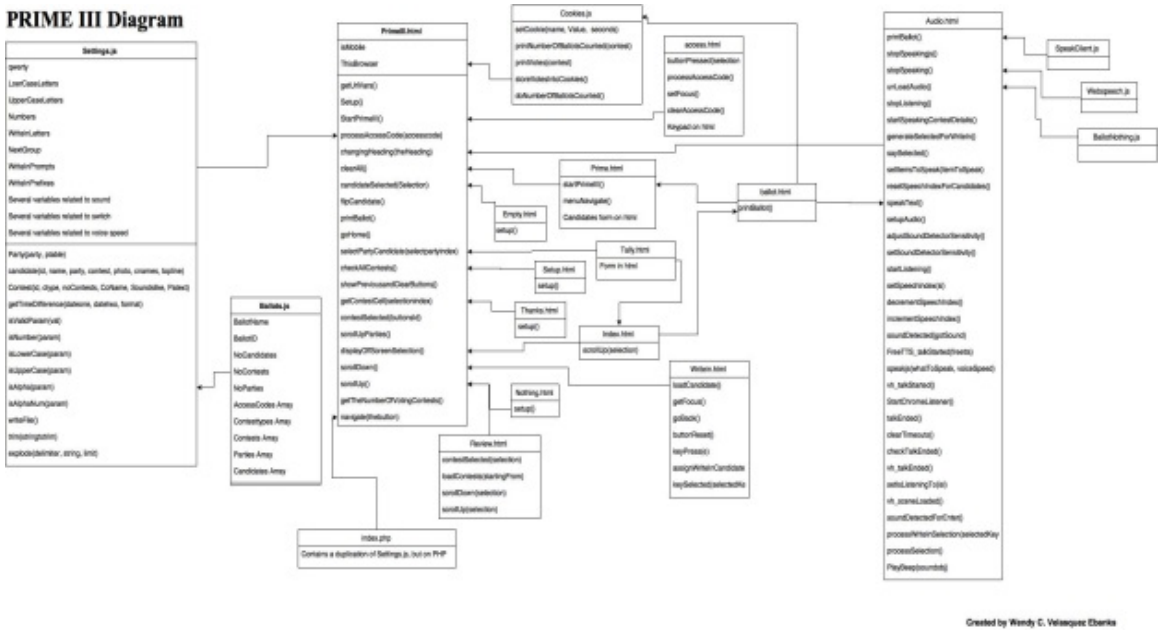

Figure 1: Prime III Diagram Structure

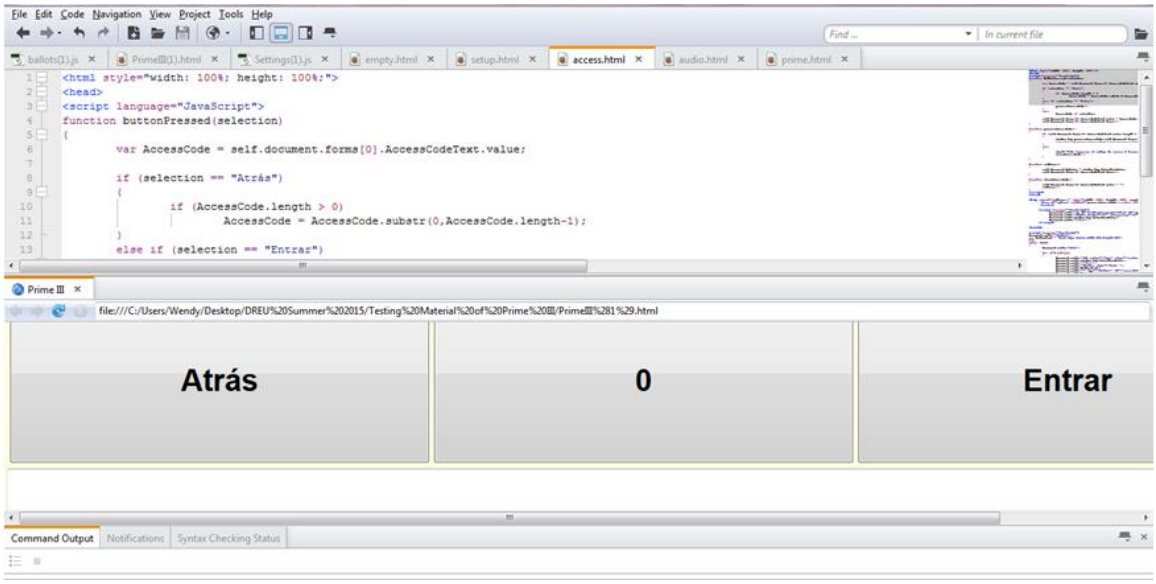

Figure 2: Manual Translation Inside of Prime III Code

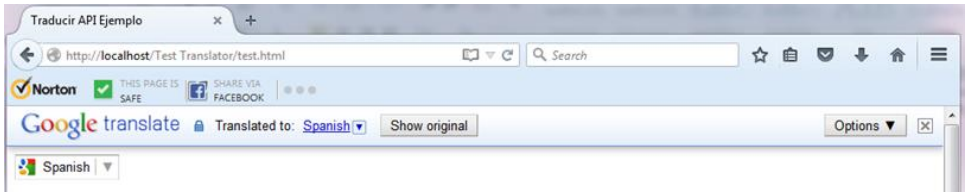

Figure 3: Selector from Google Snippet Code

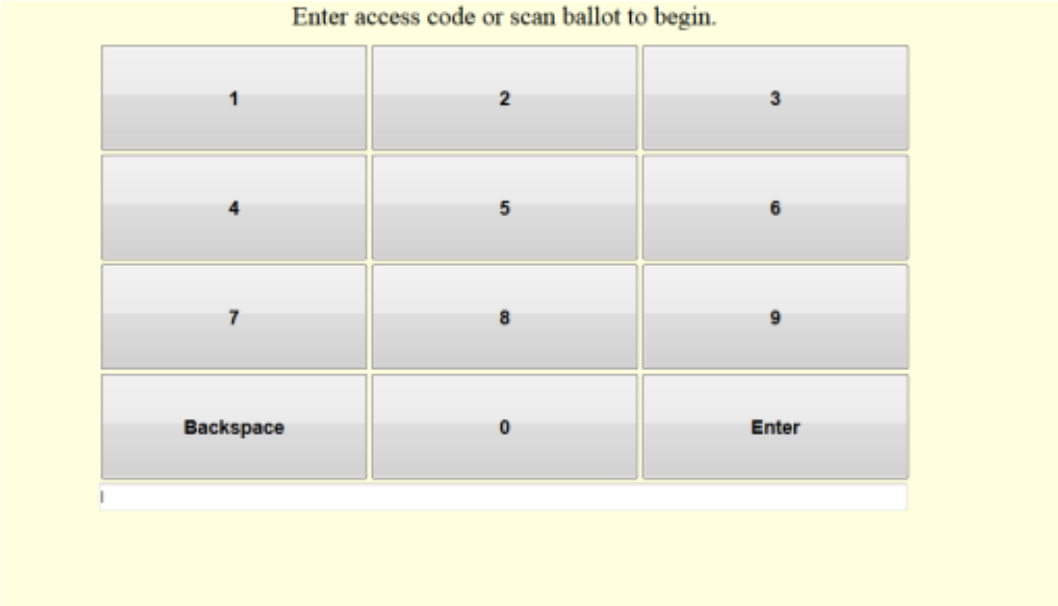

Figure 4: Prime III Original Interface

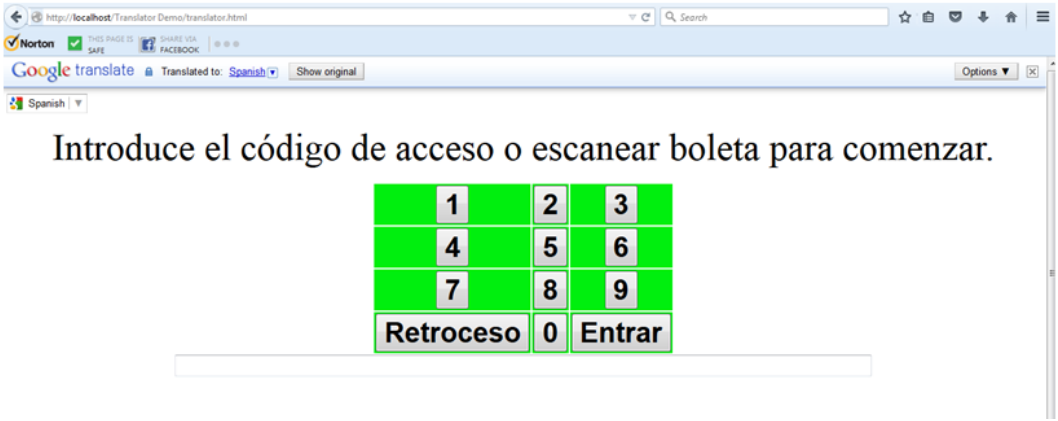

Figure 5: After Using Google Snippet Code

```
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <title>Translator</title>
</head>

<body>
    <script src="https://www.google.com/jsapi?key=YOUR_GOOGLE_KEY"></script>
      <div id="languages"><p>
  <a href="?lang=en" rel="en">English</a> /
      <a href="?lang=es" rel="es">Spanish</a> /
      <a href="?lang=it" rel="it">Italian</a> /
      <a href="?lang=fr" rel="fr">French</a>
</p></div>

    <div id="languageBlock">
```

Figure 6: Access Key Required from Google

API's	Advantages	Disadvantages
Google API using DEMO translator code	(+) It has been proved on the demo	(-) it can be used only with Chrome not any other browser. (-) Demo Code differs from the demo template (-) It needs to be connected to the internet in order to work
Google Snippet Code from Site Manager	(+) It changes generically to any lenguage indicated based on the languages Google has available for translation	(-) Changes PRIME III interface. (-) Translation is not always accurate particularly some words in the context needed it (-) code cannot be modified. (-) uses online libraries., as a result it depends from the internet connection
Google API using Access Key	(+) Allows the user to create and manipulate variables for language as needed	(-) Requires a paid Access Key in order to be used and modified. (-) It needs internet connection.
Microsoft Translate API	(+) Provides the possibility to manipulate functions related to translation	(-) the programmer need to have an access token to use and manipulate it. (-) https://msdn.microsoft.com/en-us/library/hh454950.aspx

Figure 7: API's comparisons

Figure 8: Object structure Draft

Default Language (English)	Spanish Equivalence
Back	Atrás
Enter	Entrar
Enter Access Code or Scan Ballot	Introduzca el Código de Acceso o Escanee la Boleta
You must enter an access code or scan a barcode to begin.	Para poder iniciar usted debe introducir in código de acceso o escanear la boleta
Invalid access code or barcode. Please try again.	El código de acceso o código de barra es inválido. Por Favor intente de nuevo.
Please Wait...	Por Favor Epere...
Are you sure you want to clear the tally?	¿Está seguro que desea borrar la cuenta?
Settings	Propiedades
Top	Arriba
Clear	Borrar
Review	Revisar
Continue	Continuar
Very Fast	Muy Rápido
Fast	Rápido
Average	Regular
Slow	Despacio
Very Slow	Muy Despacio
NIST Standard Ballot	Boleta estándar NIST
Vote By Party	Votar por pártido
To start voting touch selection.	Toque una selección para empezar a votar.
To start voting, touch a selection on the left.	Para iniciar Votación, toque una selección a la izquierda.
Start Voting	Empezar a Votar
Contest	Concurso
President and Vice-President	Presidente y Vice-Presidente
US Senate	Candidato a Senador de E.U.A.
US Representative	Representate a la camara del Congreso de E.U.A.
Governor	Gobernador
Liutenant-Governor	Vice-Gobernador

Figure 9: Glossary of Phrases

ASCII Dec	ASCII Hex	Symbol	HTML Number	HTML Name	
224	E0	à	à	à	latin small letter a with grave
225	E1	á	á	á	latin small letter a with acute
226	E2	â	â	â	latin small letter a with circumflex
227	E3	ã	ã	ã	latin small letter a with tilde
228	E4	ä	ä	ä	latin small letter a with diaeresis
229	E5	å	å	å	latin small letter a with ring above
230	E6	æ	æ	æ	latin small letter ae
231	E7	ç	ç	ç	latin small letter c with cedilla
232	E8	è	è	è	latin small letter e with grave
233	E9	é	é	é	latin small letter e with acute
234	EA	ê	ê	ê	latin small letter e with circumflex
235	EB	ë	ë	ë	latin small letter e with diaeresis
236	EC	ì	ì	ì	latin small letter i with grave
237	ED	í	í	í	latin small letter i with acute
238	EE	î	î	î	latin small letter i with circumflex
239	EF	ï	ï	ï	latin small letter i with diaeresis

ASCII Dec	ASCII Hex	Symbol	HTML Number	HTML Name	
240	F0	ð	ð	ð	latin small letter eth
241	F1	ñ	ñ	ñ	latin small letter n with tilde
242	F2	ò	ò	ò	latin small letter o with grave
243	F3	ó	ó	ó	latin small letter o with acute
244	F4	ô	ô	ô	latin small letter o with circumflex
245	F5	õ	õ	õ	latin small letter o with tilde
246	F6	ö	ö	ö	latin small letter o with diaeresis
247	F7	÷	÷	÷	division sign
248	F8	ø	ø	ø	latin small letter o with slash
249	F9	ù	ù	ù	latin small letter u with grave
250	FA	ú	ú	ú	latin small letter u with acute
251	FB	û	û	û	latin small letter u with circumflex
252	FC	ü	ü	ü	latin small letter u with diaeresis
253	FD	ý	ý	ý	latin small letter y with acute
254	FE	þ	þ	þ	latin small letter thorn
255	FF	ÿ	ÿ	ÿ	latin small letter y with diaeresis

Figure 10: HTML ASCII codes