

TRANS_VERBAL_ATOM

Group 10

May 3, 2004

Professor Man

Neesha Blake

Roy Guzman

Pearl Odu

Cecilia Ogude

Joyce Reyes

I pledge my honor that I have abided by Stevens Honor System.

Table of Contents

I. Abstract.....	2
I.1 Acknowledgment.....	3
II Project Final Design Plan	
II.1 Introduction.....	4
II.2 Design Requirements.....	5
II.3 System Design.....	
.....	9
II.4 Financial Budget.....	11
II.5 Project Schedule.....	12
III Funding.....	13
IV Summary.....	14
V References.....	15
VI Appendix.....	16

I Abstract

This report is written with the intention of providing the reader with a general overview of the design, construction and implementation of the Trans_verbal_ator. With this report we wish to show aspects of the design of the Trans_verbal_ator that constitute it's basic machinery; the programming involved in the coding for the Trans_verbal_ator, the software utilized to accompany the Trans_verbal_ator device and most importantly the linking of these two aspects to create the final desired output.

The intention behind the Trans_verbal_ator is to allow for communication, communication across long distances and at the same time communication over the barrier of language. With this in mind the Trans_verbal_ator is designed using programming algorithms

that allow two users to connect with each other over the internet. The device also allows for the users to speak (or type) in a message in their own respective languages while the other user on the opposite end of this connection sees the sent message in his or her own language. This area of the device utilizes the software that will (along with the coding for the chat capability) have to be linked with each other to allow for a seamless flow of information from one user to the other. As one would think the foundation of the correct operation of the Trans_verbal_ator is based on this “linking”. If we can create the device so that all aspects from the voice intake to the text display to the transfer over the internet and finally to the ultimate display of the message at the other end of the connection is performed without glitches and as quickly as possible, we can say that we have communicated two users successfully and efficiently. Also, one can say that the Trans_verbal_ator has performed it’s task.

I.1 Acknowledgements

We would like to thank the help and advice we have received from our Senior Design Advisor Professor Man without whom the idea of the Trans_verbal_ator would not have come to fruition and who had and has continuously provided information and aid whenever it was needed. Also, we would like to thank Rick Ellis from NextUp for agreeing to provide us with the textAloud COM API for free. We would also like to thank David Silverberg who gave us contact information of others that might be interested in helping with the Trans_verbal_ator. A huge amount of thanks is due to Ricardo Arguello and Eric of Word Magic Soft who created and supplied us with a web server to help with translation for the Trans_verbal_ator. And last but not the least we are also in deep gratitude to all those that have aided in making the Trans_verbal_ator move from only a concept to the point at which it is now a reality.

II.1 Introduction

Communication is and has always been an important aspect of life. With the growing technological advances and the growing need for these technologies, communication has still been able to withstand the hands of time. We are now more than ever able to communicate with people over incredible distances and in instantaneous periods of time. Considering how one can communicate as far and as fast as possible, why not make it possible to communicate to whomever possible. That is, make it possible for an English speaking American to communicate with a Spanish speaking person from his or her respective country overcoming the boundaries of space, time and now, language. The Trans-verbal-ator will do exactly this.

The Trans_verbal_ator is a sophisticated compilation of different software put together to create a device that makes it possible for two people of different tongues to communicate with each other both visually and audibly while comfortably speaking in their respective languages.

Originally the idea behind the Trans_verbal_ator was to allow the two users communicate

with each other while they typed in their messages in their respective languages. Now, we have added the ability for the user to speak into a microphone and display his spoken words as text after which he can send this text across the server to the user on the other end. Other things we might add to the device may be visual communication abilities, such as installing viewing windows to allow users with video cameras send their real time video frames across the server. This and other aspects that might add to the aesthetic nature of the Trans_verbal_ator might be supplemented to the device depending on time and energy constraints.

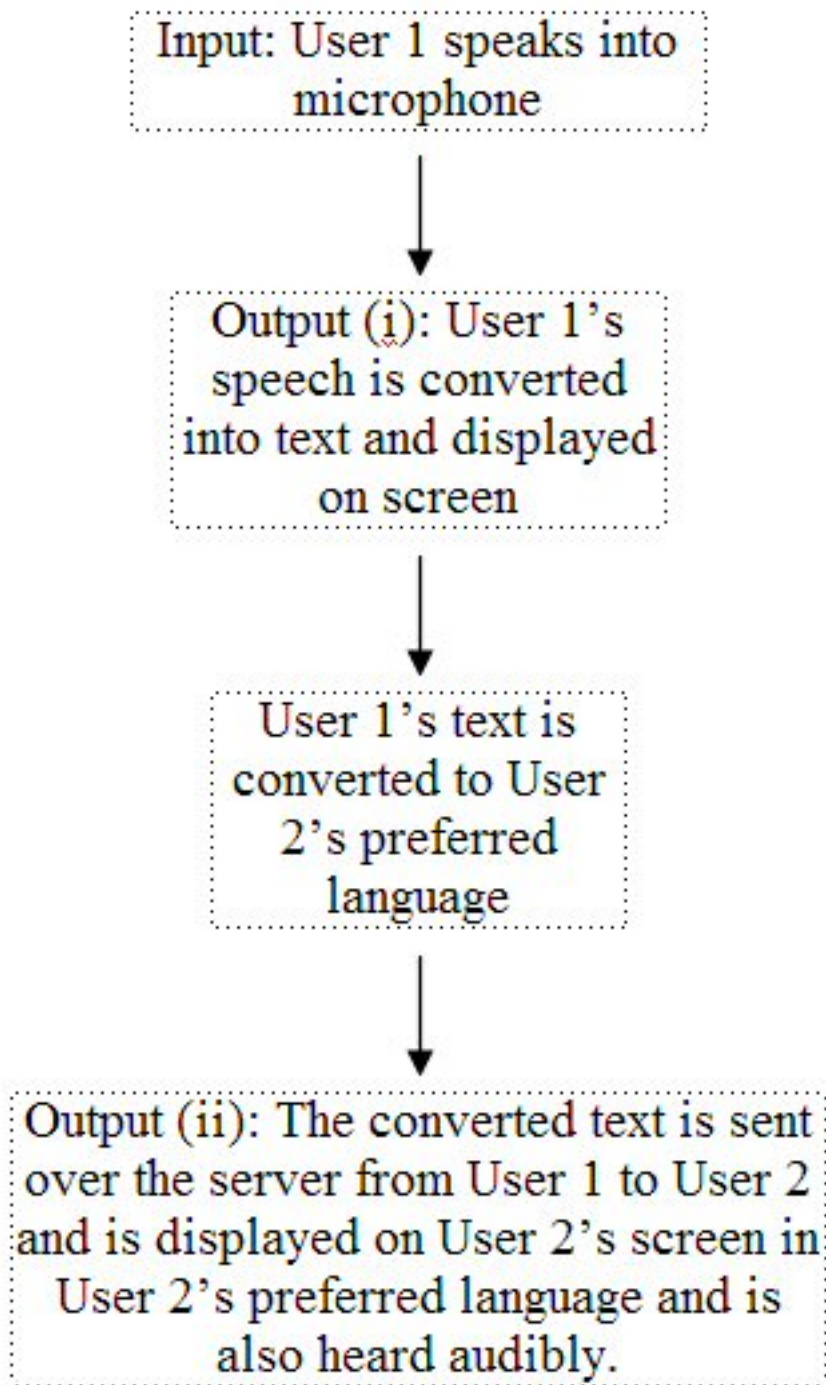
With efficient programming methods and accurate software technology the Trans_verbal_ator should allow for the flawless flow of information from one user to the other. The programming methods used in the design of the device are from regular programming codes such as JAVA and C++ including some visual basic coding for the interface of the program for it's appearance on screen. The actual linking of the different purchased software together with these programming methods is the arduous task that is the breadboard of the Trans_verbal_ator's design.

Our task was to create this device utilizing the theoretical skills we have learnt from the past years as students of Stevens Institute of Technology and New York University as well as the practical skills we gathered from experience in our college careers. The group had to accomplish this task ensuring that we were within a prescribed budget and an allotted time frame.

In this report we hope to deliver a basic idea of the technical actions we suggest are relevant and intend to implement in the fabrication of our device as well as the time and money it would take for such a device to be fabricated.

II.2 Design Requirements

As has been explained in previous sections of this proposal, in order for the Trans-verbal-ator to perform it's prescribed task it will have to have the ability to convert speech-to-text, then convert that text to the appropriate language representation, and then send that through a client-server connection to another computer and then perform a final text-to-speech conversion. The diagram below describes the procedures the Trans-verbal-ator should take for one cycle of operation.



The same process described above is repeated for User 2 with User 2's voice intake as the input and User 1 now on the receiving end of the interaction.

In order for the events shown above to occur properly and efficiently certain specifications have to be met by the Trans-verbal-ator. It would have to perform accurate speech-to-text and text-to-speech conversions, it would have to be able to translate accurately from one language to another without losing most of the "gist" of the message and it would have to be able to send this information promptly across a server to another user.

Accurate Speech-To-Text and Text-To-Speech conversion:

Speech to text has been an on-going endeavor for engineering and computer scientists and till today has not been perfected. For accurate speech-to-text conversion the minute changes of decibels in one's voice from one person to another would have to be accounted for as well as the numerous languages one can choose to speak in. Different people with their respective accents and respective tones all trying to say the same word all have the possibility of saying that one word differently. This is the voice recognition area of speech-to-text which seems to be the largest hurdle to leap over. After voice-recognition is performed what remains is matching the "voice-recognized" word to its equivalent in text format. Another tedious process considering there are over 100,000 words that need to be identified, mapped and displayed and depending on which language you choose to use this number increases. Considering all these obstacles in the way of one performing precise speech-to-text recognition and the amount of time it has taken a lot of engineers and scientists to come up with what we have out now in the market, we decided to use an already existing speech-to-text converting software and then try from there to figure out ways to manipulate it for our own device. In this way we are able to provide with the Trans-verbal-ator a device that has the most accurate as-of-the-moment voice recognition capabilities. As this goes for speech-to-text the same goes for text-to-speech. The numerous amounts of words and the time it would take to recognize each word and its corresponding pronunciation in its corresponding language as well as the engineering time that would be necessary would be far longer than the time we have available for the fabrication of the device thus, like before we have decided to use a speech-to-text software which like the former are available in different languages. In terms of languages for the moment we are concentrating on the trans-verbal-ator being able to work between two languages; English and Spanish implying that we would have need for two speech-to-text software applications and two text-to-speech software applications (each one in English and Spanish).

Accurate language translation:

Considering that we wish for users of two different languages to communicate, it is very important that they understand each other. That is to say that during an interaction with the device, we wish that important information from one user is not lost before it reaches the other user. In order to achieve this there has to be close to accurate language translation between the two languages. For this purpose because we are working with devices that perform rather precise recognition of Speech-to-text thereby implying a rather large amount of words available for translation we decided to use a translation software which in turn would have a word bank close to or as much as that of the voice recognition device. Using this method we decrease the time it would take for us to gather, store and match all these words as well as possibly the inaccuracy and errors of us trying to perform the grammatical aspect of the conversion ourselves. The software would also have to be able to allow for minor manipulations since we intend to

implement it in the Java code. The entire process of the language translation part of the trans-verbal-ator's operation would have to occur at the same time as that of sending the information over the server. This would mean it would have to occur almost instantaneously and in the background of the trans-verbal-ator's operation. Giving us another reason for the necessity of a good software application as well as its ability to be manipulated to work in different environments.

Client-Server chat capability:

We wish for the Trans-verbal-ator to be able to allow for two users to communicate with each other through a server. We intend to setup a server that would allow for clients (the Users) to connect themselves to and in turn allow for the clients to connect with each other. There are different kinds of programming methods that are available for us to use in order to achieve the goal set above. We intend to use Java programming with the SQL API (Application Programming Interface). The basic idea behind programming Java with the SQL API for chat rooms is that it enables one to start a connection and then allows a user to store their information using databases on the server and then sends this information to the user who requests it or who is open to a request for information. Using this method we should be able to manipulate the information received from the user before it is sent to another user (since we intend to translate whatever User 1 said in one language into User 2's language). In terms of manipulating the different aspects of the Trans-verbal-ator; Speech-to-text, text-to-speech and language translation so that they can work with the client-server and allow two users to send and receive information we intend to let each process occur step by step. That is to say, we wish that once the information is converted into text from speech it can then be converted into the respective language, then that information can be sent to the database which is where we use the text-to-speech software and at the same time send the information over to the other user so they can view what the other user wrote but now in the language they prefer.

The "linking up" problem:

As has been explained in earlier paragraphs "linking up" is the biggest issue one encounters in designing a device such as the Trans_verbal_ator that operates based on systematic relationships between other devices. The linking up problem has to some extent given limitations on the original capacity of the Trans_verbal_ator. For example, with most text-to-speech software requiring that the text be pasted in the window of the software application the grabbing of text from the chat window to the text-to-speech application window and then hitting the "speak" button to hear the text all to be done using nothing but code seems a little implausible. Although there might be a method of coding that will allow for such a thing to be done our time constraints and the limited scope of knowledge we know with programming inhibit us. This aspect - allowing user2 to hear the translated text sent from user1 seems to be something we

might put off until we have time and the skill to attach it as an add-on to the device.

Another linking-up issue is that of the speech-to-text then to the chat window. This area unlike the area just described above seems to be a little bit more feasible. With the technology available speech to text can be performed to convert the sound input and display the text right into windows-based programs (such as Microsoft Word) and because we intend to create the application to work on a Windows platform it should not be a difficult problem to write code that grabs text from a saved file and prints it out in a chat window. Such programs have already been written before.

The translation linking-up issue is a bit similar to that of the text-to-speech but in this case we simply have to find a particular program that allows us translate the information by placing the text in the translator application and then copy the translated text from there and place it in the chat window. All that is needed is a code that performs exactly this. If the application has the ability of converting the translated text automatically into a text file copying and pasting from that text file should be easy to do with code.

Hence, one can see that the bulk of the project is concentrated in creating algorithms and code so that make it possible to successfully link all aspects of the Trans-verbal-ator concisely and efficiently so in the end the Trans-verbal-ator, as described in the introduction will perform to the best of its ability its prescribed tasks.

II.3 System Design

Our project, the trans-verbal-ator, consists of many components that came together to create an application that will help in eventually eliminating language barriers over any network. Like every application there were some software aspects as well as hardware components. First and foremost, our project needed to convert anything the users say into text that the software can readily translate. The software had to be able to recognize the word(s) being said by the user and convert it to text, which would then be later translated into the text version of the other language. The group was able to obtain a software program that would capture the speech and display it as text on the screen. This program is called dragon naturally speaking 7.0. Fortunately, there are two versions for English and for Spanish. This package brings a headset along with the hardware that gives this product its voice recognition capabilities and the program that shows the text output of the voice input.

Originally, we wished for our Software to also have a text to audio component. For this part no hardware would have been necessary, since every computer already has this capability of producing sound. It would all just require some programming or using any software out in the market that could already do this. The group came across a product called textAloud. This product actually converted text according to pronunciation of the text. Our application could then be able to convert this text into the other language. We however, because of funds and time

constraints were unable to implement this aspect of the Trans_verbal_ator's functionality into effect. We ended up staying rather with the English-speaking user being able to talk into a microphone and the Spanish-speaking user being able to view the converted text. For the translating aspect of the project we were originally going to work with using "if" and "else" statements within in our programming environment but wished that we could make the Trans_verbal_ator as efficient as possible. Thus, we began searching for already in-the-market translation software that we could use in some way with our program. After searching for quite a while and communicating quite a number of people we stumbled across Word Magic Soft. Word Magic Soft is a translation tool that one can download to their computer. The efficiency of this program as a translation software is remarkable in that not only could it translate regular English phrases that you typed into the window and Spanish to English phrases as well, it could also translate slang. This was a feature perfectly fitting for a messaging service like the Trans_verbal_ator. After contacting and talking through email and by phone constantly, Ricardo Arguello and Eric of Word Magic Soft were able to create a server specifically dedicated to us and the Trans_verbal_ator. With this server we could simply just implement a code in our program that would direct the inputted text from the English or Spanish user to be sent to that web-server, translated and then sent back to the Trans_verbal_ator window. This way, we were having the fast communication and the best translation together in one.

Before we could implement the translation aspect we had to first create a user interface such as a chat room or an instant messaging server to connect two or more users. Of course, each user would have had to have the software installed to have audio and text capabilities of the program. The final interface of our program can be seen in figure A1 in the appendix. Part of the programming code can be seen on the next page in appendix A2. After creating the chat program in JAVA we opted to re-create the program in a different programming language, especially one that had good interfacing capabilities as well. Our final choice came to Visual Basic. After researching on the internet and in books on the programming software we ended up obtaining a very sufficient internet messaging program that used Winsock for it's client-server capability. With manipulation of the program and addition of the coding to connect to the server created for us by Word Magic Soft we were able to create the Trans_verbal_ator so it had good translation capabilities while at the same time being pleasing to the eye.

II.4 Financial Budget

Below we have provided a table showing our estimated costs for the weeks that we spent on the project with 5 team members. On average, each team member spends 6 hours each week on the project. The table shows our estimated costs for direct labor, materials and parts, software, test equipments, documentation costs, miscellaneous expenses like phone charges to vendors, and traveling expenses. These costs include costs that were incurred from the design inception a

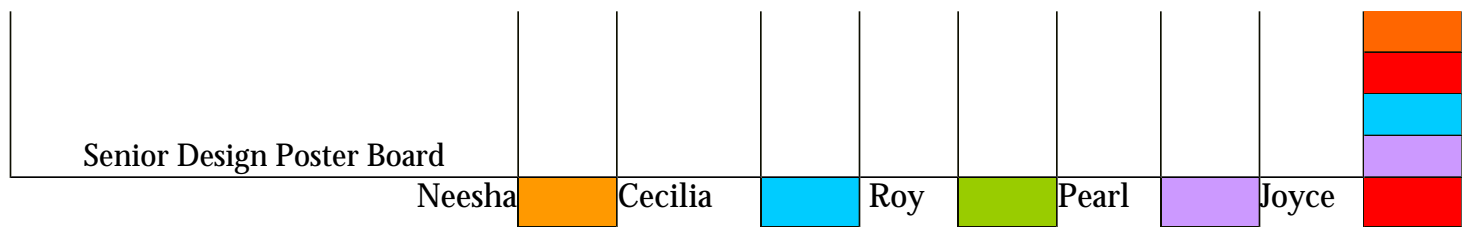
semester ago to the final output we achieved this semester.

Components	Description	Quantity	Price	Total
Direct Labor	Estimated 470 hours	5	\$19/hr.	\$8,930
Materials/Parts	Compaq Armada E500	2	\$1,200	\$2,400
	Audio Technical Unidirectional Voice/ Entrustment Microphones	1	\$19.99	\$19.99
Software	Microsoft Visual Basic C++	1	\$109.00	\$109.00
	Dragon NaturallySpeaking SDK	1	\$695.00	\$695.00
	WorkMagicSoft Server	0	\$0.00	\$0.00
	Adobe Photoshop	1	\$649.00	\$649.00
Test Equipment	Lab supplied by school		0	0
Documentation Costs	Printing Brochures	30	\$0.83	\$30.00
	Printing Posters	2	0	\$0.00
Miscellaneous Expenses	Phone charges to vendors		\$20	\$20.00
Traveling Costs	Traveling to and from Hoboken and NYC		\$9/wk	\$117
Total				\$12,970

II.5 Project Schedule

II.2 Project Schedule	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Project Task									
Identify Design Concept									
Research									
Visual Basic, C++, and Java									
Client Server Systems									

Company Sponsors	[Purple Bar]				
Reformulate Design Concept	[Cyan Bar]				
	[Orange Bar]				
	[Purple Bar]				
	[Green Bar]				
	[Red Bar]				
Economics Budget	[Cyan Bar]				
Senior Design Webpage	[Orange Bar]				
	[Red Bar]				
	[Purple Bar]				
Project Proposal	[Cyan Bar]				
	[Orange Bar]				
	[Purple Bar]				
	[Green Bar]				
	[Red Bar]				
Construction and Design		[Purple Bar]			
		[Green Bar]			
		[Red Bar]			
Record Spanish/English		[Purple Bar]			
		[Green Bar]			
		[Red Bar]			
		[Orange Bar]			
		[Cyan Bar]			
Final Project Report		[Orange Bar]			
		[Green Bar]			
		[Cyan Bar]			
		[Purple Bar]			
		[Red Bar]			
Java Programming		[Purple Bar]			
		[Green Bar]			
Speech Recognition Trial			[Cyan Bar]		
			[Orange Bar]		
			[Red Bar]		
			[Purple Bar]		
			[Green Bar]		
Interim Report				[Orange Bar]	
				[Green Bar]	
				[Cyan Bar]	
				[Purple Bar]	
Design Brochure				[Red Bar]	
				[Cyan Bar]	
				[Purple Bar]	
				[Orange Bar]	
				[Red Bar]	
Contacting WordMagic Soft				[Green Bar]	
				[Cyan Bar]	



II.6 Funding

Our group contacted all of the companies whose software we thought would be necessary for the Trans_verbal_ator's functionality.

Dragon NaturallySpeaking does not provide any free software. However, the company was kind enough to suggest website that sells the product for a cheaper price. At <http://www.next-wave-solutions.com/dragnatprof.html>, Dragon NaturallySpeaking sells for \$593.95 instead of \$695. We would still need to buy the SDK for the software.

Word Magic Soft was incredibly helpful. Having their device available to us on the web-server was an amazing help. We are very much indebted to Richard and Eric for their help.

III Summary

Finally, after extensive research on the different components that make up the Trans_verbal_ator and serious considerations on how to manipulate these components so they work together to perform the actions required by the Trans-verbal-ator: the ability to allow two people communicate comfortably and efficiently while speaking their own respective languages we are grateful that our ideas were realized and our goals achieved.

This project was an amazing an experience to go through with and we believe it will help to revolutionize communication in the social level, cooperate level as well as the government sector. We hope the Trans-verbal-ator will end up being a wanted commodity.

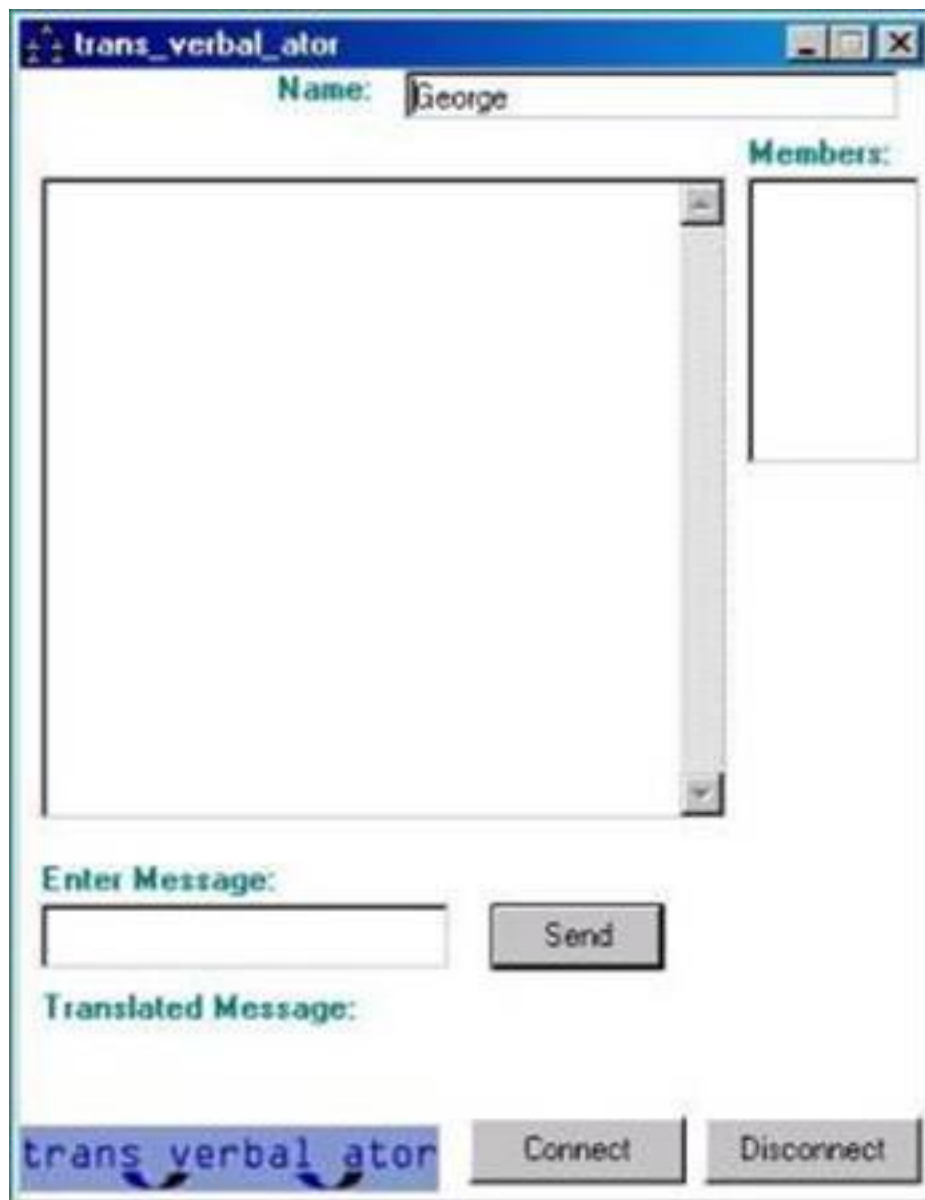
IV References

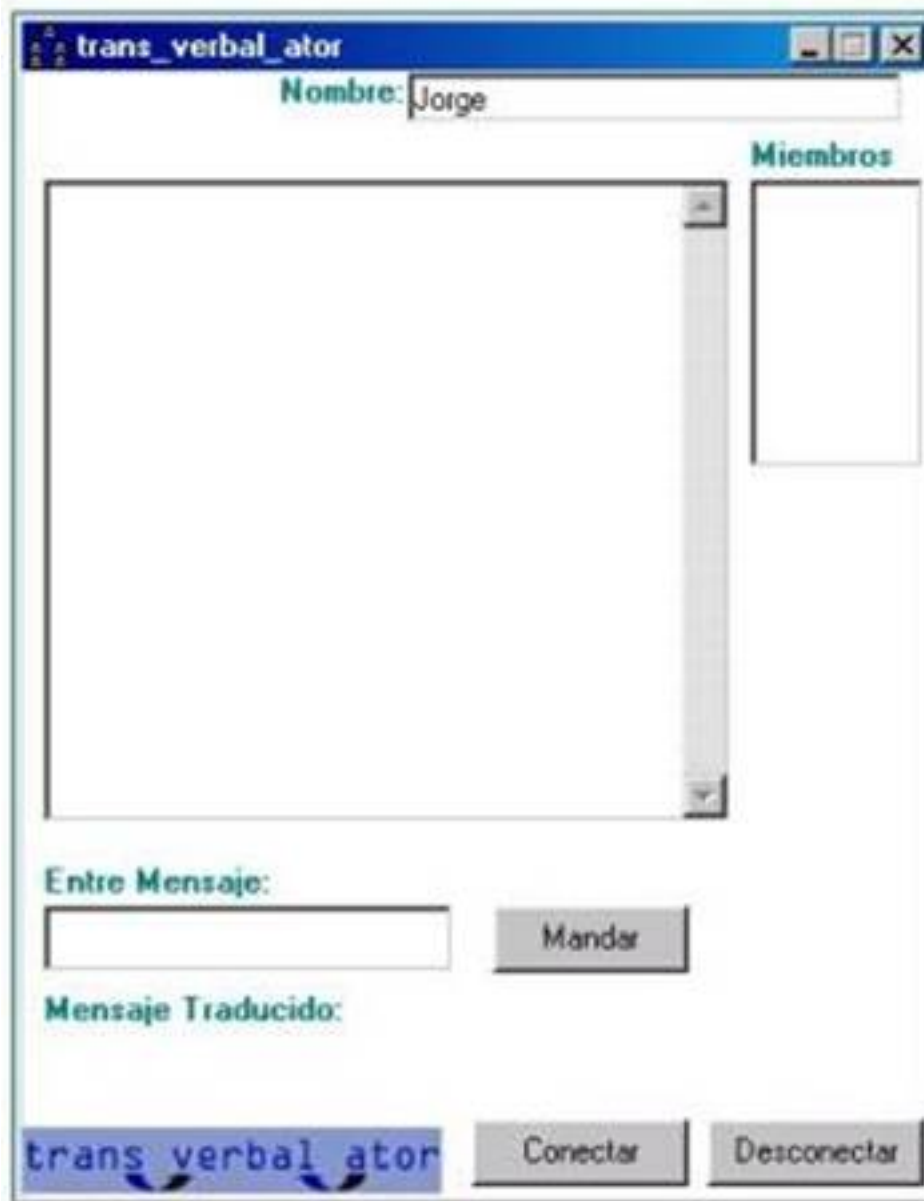
1. Circuit City <<http://www.circuitcity.com/init.jsp>>
2. Dragon NatuallySpeaking 7 <<http://www.scansoft.com/naturallyspeaking/>>
3. Electrical and Computer Engineering Senior Design Website <<http://koala.ece.stevens-tech.edu/sd/>>
4. Dragon Systems. <<http://www.dragonsys.com>>
5. Free Translation <<http://ets.freetranslation.com>>

6. Google Search <<http://www.google.com/>>
7. Bloch, Joshua. (2001) Effective Java Programming Language Guide [Addison-Wesley]
8. MySQL <<http://www.mysql.com>>
9. Next Up <<http://www.NextUp.com>>
10. Brown, Steven. (1998) Visual Basic 6 Complete [Sybex]
11. Synapse Adaptive <http://www.synapseadaptive.com/naturallyspeaking/developer.htm>
12. Textaloud <http://www.textaloud.com>
13. United Nations <<http://www.un.org>>
14. Word Magic Soft <<http://www.wordmagicsoft.com>>

V Appendix

A1





A2

Option Explicit

Public LoginSucceeded As Boolean

Private Sub cmdCancel_Click()

 'set the global var to false

 'to denote a failed login

 LoginSucceeded = False

 Me.Hide

End Sub

Private Sub cmdOK_Click()

```
LoginSucceeded = True
'check for correct password
Me.Hide
frmMain.Show
```

End Sub

For User 1, the English speaking user:

Private Sub cmdSend_Click()

'Someone clicked the Send button to send a message.

On Error Resume Next

```
'Call the wordsoft service website
Dim wm As New wmlibservice.libServiceClass

enteredText = wm.TranslatePhrase(True, txtMessage.Text)
```

For User 2, the Spanish speaking user:

Private Sub cmdSend_Click()

'Someone clicked the Send button to send a message.

On Error Resume Next

```
'Call the wordsoft service website
Dim wm As New wmlibservice.libServiceClass

enteredText = wm.TranslatePhrase(False, txtMessage.Text)
```

Some If-else statements that were to be used in the original programming of the translate method:

Public Sub translate()

```
enteredText = txtMessage.Text
If txtMessage.Text = "hello" Then enteredText = "hola"
If txtMessage.Text = "i'm hungry" Then enteredText = "tengo hambre"
If txtMessage.Text = "thank you" Then enteredText = "Gracias"
If txtMessage.Text = "excuse me!" Then enteredText = "¡Oiga, por favor!"
If txtMessage.Text = "goodbye!" Then enteredText = "¡Adiós!"
If txtMessage.Text = "how are you?" Then enteredText = "¿Cómo está?"
If txtMessage.Text = "fine, thanks" Then enteredText = "Bien, gracias"
If txtMessage.Text = "and you?" Then enteredText = "¿Y usted?"
```