Project 2002-03

SPEECH ENABLED BROWSER

Estd-1984

ISO 9001:2000 Certified

Project Report

A Study by

Vinay Mohan
III B.Sc - Computer Technology

Under the guidance of

Mrs. V. Geetha, M.C.A
Senior Lecturer,
Department of Computer Technology

In partial fulfillment of the requirements for the award of degree of

Bachelor of Science - Applied Sciences
Computer Technology

Of Bharathiar University, Coimbatore - 641 046

DEPARTMENT OF COMPUTER TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

Estd-1984

ISO 9001:2000 Certified

Department of Computer Technology

Certificate

This is to certify that this project entitled

SPEECH ENABLED BROWSER

has been submitted by

Mr. VINAY MOHAN

In partial fulfillment of the requirements for the award of degree of Bachelor of Science - Applied Sciences, Computer Technology of Bharathiar University, Coimbatore:641 046 during the academic year 2002-2003.

2/13

(Project Guide) **(Head Of the Department)**

Certified that the Candidate was Examined by us in the Project Work Viva-Voce Examination held on 24-03-2003

University Register Number: 0028Q0171

2/4/13

(Internal Examiner) **(External Examiner)**

"There was a time in the world when too much talking could put a person in deep trouble....

....Today, in the age of intelligent machines, it can, literally, take us places."

Contents

ACKNOWLEDGEMENTS

Acknowledgements

If this project goes down as a success, then it would be the result of a very hard six months of persistent efforts. And if not for the help and advice of many people, this project would not have been what it is today.

At the outset I would like to acknowledge the efforts of our principal, Dr. K. K Padmanabhan and the Head of our Department, Prof. V .Sundaram who have provided us with excellent support at all levels in our collegiate life.

My deepest sense of gratitude goes to my Project Guide, Mrs. V. Geetha who has been an exceptional motivator. It is with the freedom and encouragement that she gave me during the project that I have been able to perform, even under times of academic pressure. Mrs. Geetha's appreciation for my work and constant support has helped me stand by tough times.

I am indebted to Mr. V. Sivakumar and Ms. Ubellah Maria, members of our department faculty who gave me novel ideas at every stage. The main idea behind this project was Mr. Sivakumar's brainchild and it was he who encouraged me to take it up.
My thanks to Mr. N. S. Ramalingam who helped me see rooms for improvement and gave me some thought provoking criticism.
Throughout the year, several faculty members have helped me improve this project and have shown a lot of enthusiasm in my work. My thanks go out to them too.

My heartfelt thanks to our Laboratory Technicians, Mr. Arun and Mr. Dilip, who have helped me with my presentation sessions and hardware usage.

I am deeply grateful to my parents who have been my main support throughout the years and especially during the project development stage.
My parents' difficulty in using a keyboard prompted and inspired me to develop this project!!
I am especially thankful to my brother, Mr. Shanker Krishna Kumar whose long hours on the Internet gave me access to the various software used in this project.

Finally a word of silent gratitude to all those giants of the information world on whose shoulders I stand today, as I make a place for myself with this project.

SYNOPSIS

Project 2002-03: Synopsis

The project entitled "**Speech Enabled Browser**" is aimed at developing a Browser integrated with a Speech Recognition and Synthesis Unit. The two main purposes of this integration are:

- To browse the internet using voice commands instead of keyboard text inputs.
- To receive voice outputs as feedback from web pages or on some user action taking place.

The primary features of this software are:

- A browser that can be used with manual (keyboard) inputs and voice commands.
- A Speech Recognition (SR) Unit that "listens" to commands being provided. The unit can also be shut off for manual usage.
- A translation of the voice commands into textual characters by the SR Unit.
- A Text-To-Speech (TTS) conversion unit that translates web-text into English words.
- A "History" file that will store and use the Web URL's accessed.
- Miscellaneous options to change Voices, activate Volume Controls, speak System Time and Train user voice profiles.

This browser promises to be different from others because of its ability to function as a normal browser and also facilitate browsing using Speech Technology, which is a new, popular and powerful concept.
It may not serve as efficiently as a regular browser but its uniqueness of operation and utility value make it special.

This project has been developed using Microsoft Visual Basic 6.0 and Microsoft Speech Software Development Kit (SDK) version 5.1.

PROJECT OVERVIEW

Project overview:

The end result of this project is a software that exists as a standalone .EXE file. The software is an active combination of a Browser and a Speech Recognition and Synthesis System.
The 2 main features of this integrated browser are:

- Recognition and conversion of user Voice Inputs to Web URL's.
  Example: if the user were to say "w, w, w dot hotmail dot com"
  the system will convert it to "www.hotmail.com"

- Conversion of Web Page Text to Synthesized Voice outputs.
  Example: a web page's contents can be orally read out to the user by an in-built voice.

The browser performs the regular functions like Navigation (Forward, Back, Refresh, Stop, etc).
It also handles File opening, saving, printing and loading of new windows. Options like Copy and Paste are also available.

The usage of Speech partially eliminates the usage of the keyboard. By uttering a few words, the user can browse through a network.
Also, with the help of voice feedback, there is no need to actually read the web page when it can be automatically read out. This voice is an electronically synthesized one and has a strong American accent. Male and Female voices varying in pitch and tone can be used for this feedback.

The Speech system consists of:

- The Speech Recognition Engine
- The Text-To-Speech (TTS) Engine
- The Speech objects and libraries used in the code

The Speech objects are actually used during the execution of the software. During runtime, there is a polling process that takes place between the speech objects and the engines. This resembles a Client-Server system where a set of Requests and Responses take place.

Technical System Description:

This project was developed with the following software:

Programming Language : Microsoft Visual Basic 6.0

Speech Components : Microsoft Speech SDK version 5.1

The Browser module of this software was developed using the **WebBrowser Control** of Visual Basic. This simulates the exact working of the Internet Explorer window.
Tasks like Navigation and Page access have been coded separately.

Common Dialog controls take care of File operations, Saving, Printing, etc.

The browser, as mentioned earlier, acts as an interface between the Speech System and the user. Hence, a lot of importance with respect to Graphical appearance has been given to the browser.
The browser module has a main toolbar for Navigation of web pages as in other browsers like Microsoft Internet Explorer. These tools, mainly: **BACK, FORWARD, REFRESH, HOME and STOP** are used to traverse through web pages.
The browser also has a control menu for miscellaneous options like File, Edit, Speech, Navigate and Help.
An important component of the browser is a simple Text file that acts as a History for the web pages browsed. It stores all the URL's accessed by the browser.

Other controls like Speech System Activation, Mode Switching, Volume Controls, etc have also been added. This is to control the Speech Modules which have no GUI of their own.
Each control like **Talk Mode, Type Mode, Speak, Silence** and **Pause/Resume** are placed to generate specific speech events.

The speech system works in direct communication with the Speech Engines in the Shell of the operating system. Neither the Speech system in this software nor the engines that it depends on can be actually viewed as a separate entities.
They both exist as built-in, conceptual units.

Speech API (SAPI) objects like **ISpVoice** and **SpeechRecoContext** control a set of events that are triggered every time a certain Speech object and its funtions are activated.
By inserting these triggers between the GUI and Speech event, we can obtain the results we want.

Competitor Analysis:

This software has been built upon the existence of similar systems.
In fact, browsers have been the centre of a popular software development movement including controversies because of the sheer importance that browsers carry with them.

- Microsoft Internet Explorer is arguably today's most popular browser. It's compatibility with Programming Language Scripts, Animation and Applet support make it an automatic choice.

- Close on its heels is Netscape Navigator, the original pioneering browser that brought out innovations like Status messages, progress status, scripting operations, etc.

- Another popular browser in the market today is the Opera Browser, believed to be the fastest available browser. Opera works with caches of internet data that are loaded appropriately for faster processing.

The speech enabled browser matches a few of the details from these browsers existing while it uses its own innovative speech concepts to retain a separate place in the browser world.

The general characteristics of the existing browsers are that of speed, dependency, user friendliness, high levels of control and operations for the user.
Another factor that sets these browsers apart is that they have a wide range of support for various scripting languages like VBScript, JavaScript, Cascading Style Sheets, ASP and programs like Applets, Macromedia Animation, etc.

Also, with the growth of internet technology, browser such as Internet Explorer have also begun supporting concepts like .NET, MSN messenger service and other related products and features.

The Speech Enabled Browser, however does not go deep into the power factor of browsers but has its own small place in the world of utility software where it provides new and interesting ways to access online information.

As far as originality is concerned, other Speech Enabled Browsers are yet to be developed. Until then, this software reigns supreme.

System Requirements

System Requirements Specification:

The **Software** requirements of the Speech Enabled Browser are:

- Operating System : Microsoft Windows 2000 and XP.
- Runtime Components : Microsoft Visual Basic 6.0 Runtime Files.
- Speech Components Engine : Microsoft Speech Recognition (Version 4.0/5.0) & Microsoft Speech Synthesis Engine (Version 4.0/5.0).
- Optional Software Units : Microsoft Visual Studio 6.0 & Visual Studio Service Pack 3 or later versions.

The **Hardware** requirements of the Speech Enabled Browser are:

- A Pentium II/III/IV or equivalent processor is recommended.
- Intel or Equivalent Full Duplex Motherboard.
- 32-bit sound card with Microphone and Speakers.
- System speeds of 700 Hz or greater will help in faster and better performance.
- A minimum of 128 MB RAM.
- Dedicated Hard Disk memory space of 25 MB (minimum).
- PC Modem (minimum 56.6 kbps).
- A working Telephone connection (optional; for Internet usage).

In general, a fast, internet-ready PC would be best for this software to work efficiently.

Why this Project?

Motivation behind this project:

There are several factors that prompted me to take up this project. It is based on these that I have been able to select, develop and complete this project successfully.

- The need to develop original software: to do something that had not been tried before, something for the first time.

- The need to use some latest technical concept: with the aid of a modern concept, the project has all the more value added to it. It is also made *Future-Friendly*.

- To provide some form of utility for all: the project should be usable by all; at least by a large population in the technological world.
  Usability of software is a true measure of success in development.

- To work on some existing technology: at the same time, working on some existing concept would add value to it and increase its effectiveness of use.

- To implement maximum knowledge gained: the project should be a result of all applied knowledge gained during the process of study.
  Academic and other knowledge sources must be properly used to derive maximum benefits.

The Speech Enabled Browser, was a good choice for a project, considering the factors mentioned above.
It is original (with Speech Systems), works on an existing technology (the Browser), has high usability, a novel and original idea, ready for the changing future and is the result of a lot of study and application from various corners.

The project has also presented several challenges along the way and especially in the new area of speech where lots of concepts were yet to be explored.
This project was, thus, an obvious and interesting choice.

Technical Description

Browser Technology: An overview

Browsers are User Agents that are used as Client-side software to send requests to remote software called Servers (like Apache, PWS, IIS, Tomcat, etc) over a network for some information transfer.

Browsers use a protocol called Hyper Text Transfer Protocol (HTTP) to send and receive specially encoded documents in the form of Hyper Text Markup Language (HTML) pages.

Browsers have 2 important uses:

- Act as an interface between the user and the network
- Interpret and hold the HTML documents and deliver it to the user in an appropriate manner for him to understand.

Browsers perform 3 main tasks:

- Request information transfer
- Capture and hold this transferred information
- Enable users to access web-based information in a convenient manner.

Most browsers have the following components as a user agent:

- Menu for saving and retrieving web pages
- Navigation facilities for traversing through the web
- Address holding areas for the user to explicitly specify locations
- A main window to hold the web data
- Controls to specify miscellaneous options like security, appearance, runtime issues, etc

All in all, web browsers are a unique software family that has a special place for both usage and development.

Speech Technology: An overview

This is a revolutionary concept that arose in the 1990's in close relation with Artificial Intelligence and Expert Systems.

The use of speech in computers was felt to be a big boost for technology as a whole. The difference while using a concept like speech is that a transfer of sensory control (from the eye to the ear) is made.
Users no longer have to see, but only need to listen. They need not use mechanical input devices; a few words will do the trick.

Speech Technology revolves around two inner concepts:

- Speech Recognition
- Speech Synthesis

Speech recognition is a powerful method of capturing, analysing and storing user voices for some action to be performed later. It can also be used to generate text outputs or some noticeable events and triggering of procedures.

Speech recognition, or speech-to-text, involves capturing and digitizing the sound waves, converting them to basic language units or phonemes, constructing words from phonemes, and contextually analyzing the words to ensure correct spelling for words that sound alike (such as write and right).

Recognizers-also referred to as speech recognition engines- are the software drivers that convert the acoustical signal to a digital signal and deliver recognized speech as text to your application. Most recognizers support continuous speech, meaning one can speak naturally into a microphone at the speed of most conversations.

Speech synthesis involves the artificial creation of speech. By using pattern matching and grammar recognition techniques, an actual voice can be used to "read" textual data.
This technology is truly fascinating because it emulates human emotions and behaviour like no other computing concept.

Software drivers called synthesizers, or text-to-speech voices, perform speech synthesis, handling the complexity of converting text and generating spoken language.

Although easy to understand, the voice produced by synthesis technology tends to sound less human than a voice reproduced by a digital recording.

Features of Speech Technology:

- Hands-free computing as an alternative to the keyboard, or to allow the application to be used in environments where a keyboard is impractical (e.g., small mobile devices, AutoPCs, or in mobile phones).
- A more "human" computer, one users can talk to, may make educational and entertainment applications seem more friendly and realistic.
- Voice responses to message boxes and wizard screens can easily be designed into an application.
- Streamlined access to application controls and large lists enables a user to speak any one item from a list or any command from a potentially huge set of commands without having to navigate through several dialog boxes or cascading menus.
- Speech-activated macros let a user speak a natural word or phrase rather than use the keyboard or a command to activate a macro. For example, saying "Spell check the paragraph" is easier for most users to remember than the CTRL+F5 key combination.

Potential applications of Speech Technology:

Games and Edutainment

Speech recognition offers game and edutainment developers the potential to bring their applications to a new level of play. With games, for example, traditional computer-based characters could evolve into characters that the user can actually talk to.

While speech recognition enhances the realism and fun in many computer games, it also provides a useful alternative to keyboard-based control, and voice commands provide new freedom for the user in any sort of application, from entertainment to office productivity.

Data Entry

Applications that require users to keyboard paper-based data into the computer (such as database front-ends and spreadsheets) are good candidates for a speech recognition application. Reading data directly to the computer is much easier for most users and can significantly speed up data entry.

While speech recognition technology cannot effectively be used to enter names, it can enter numbers or items selected from a small (less than 100 items) list. Some recognizers can even handle spelling fairly well. If an application has fields with mutually exclusive data types (for example, one field allows "male" or "female", another is for age, and a third is for city), the speech recognition engine can process the command and automatically determine which field to fill in.

Document Editing

This is a scenario in which one or both modes of speech recognition could be used to dramatically improve productivity. Dictation would allow users to dictate entire documents without typing. Command and control would allow users to modify formatting or change views without using the mouse or keyboard. For example, a word processor might provide commands like "bold", "italic", "change to Times New Roman font", "use bullet list text style," and "use 18 point type." A paint package might have "select eraser" or "choose a wider brush."

PROCESS DIAGRAMS

Legend to the process diagrams:

GUI : Graphical User Interface, the user interface in the form of graphical components.

Phonemes : the unitary form of data in speech system, the lowermost converted form of speech to text.

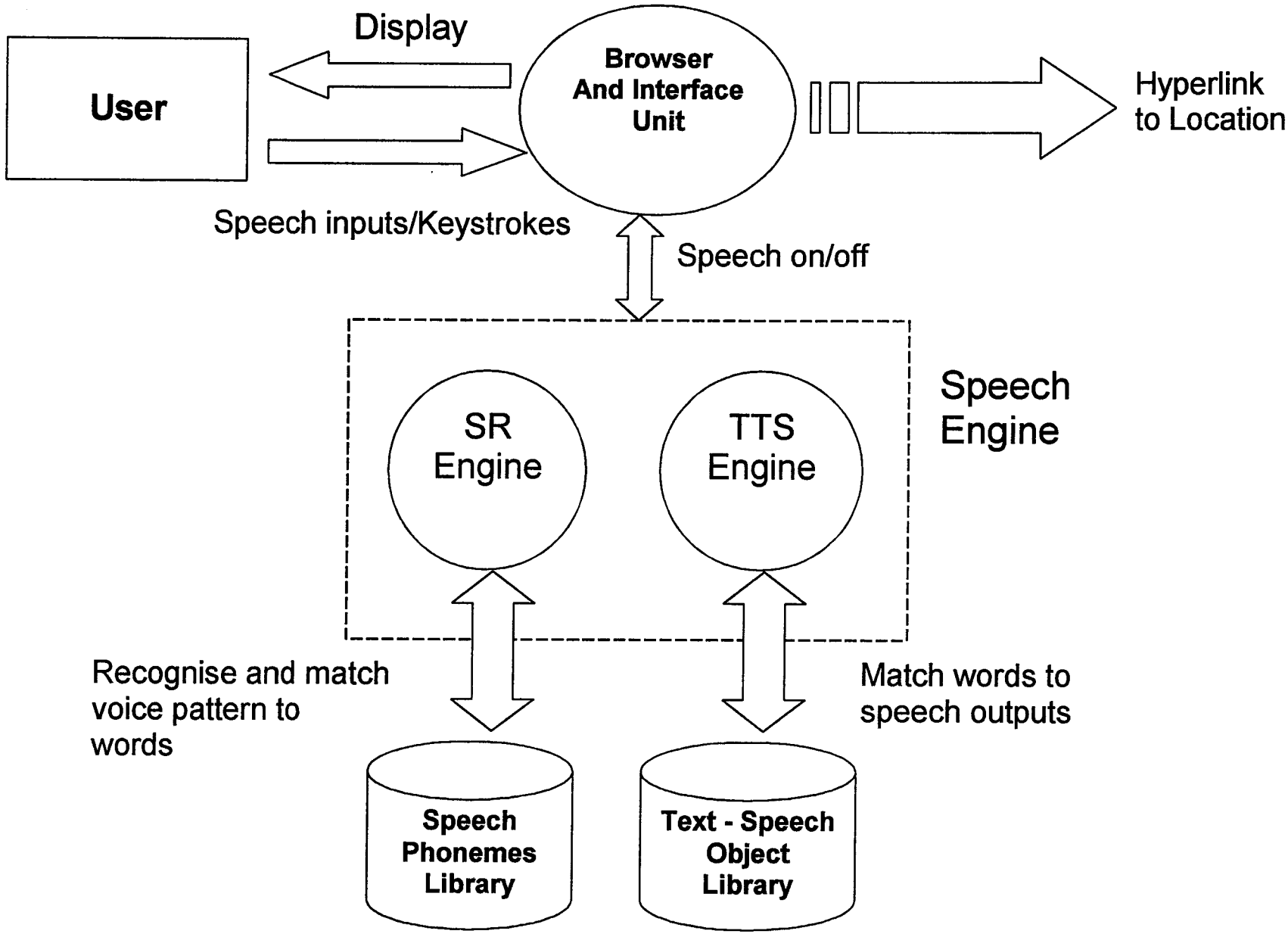
SR : Speech Recognition, the main aspect of speech technology.

TTS : Text-To-Speech, one of the 2 types of speech synthesis.

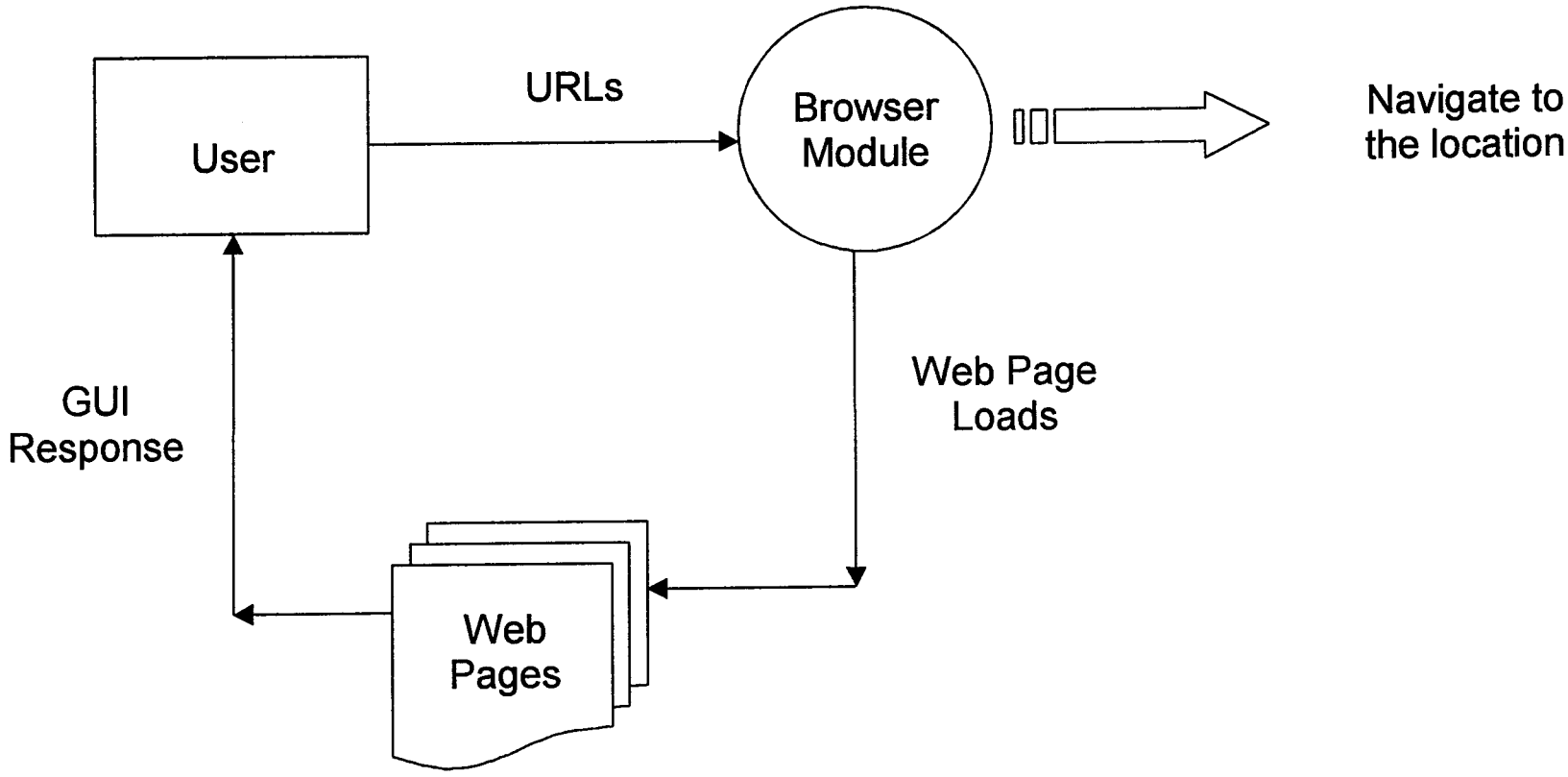
URL : Universal Resource Locator, internet addresses.

Web Text : any form of Internet text data.

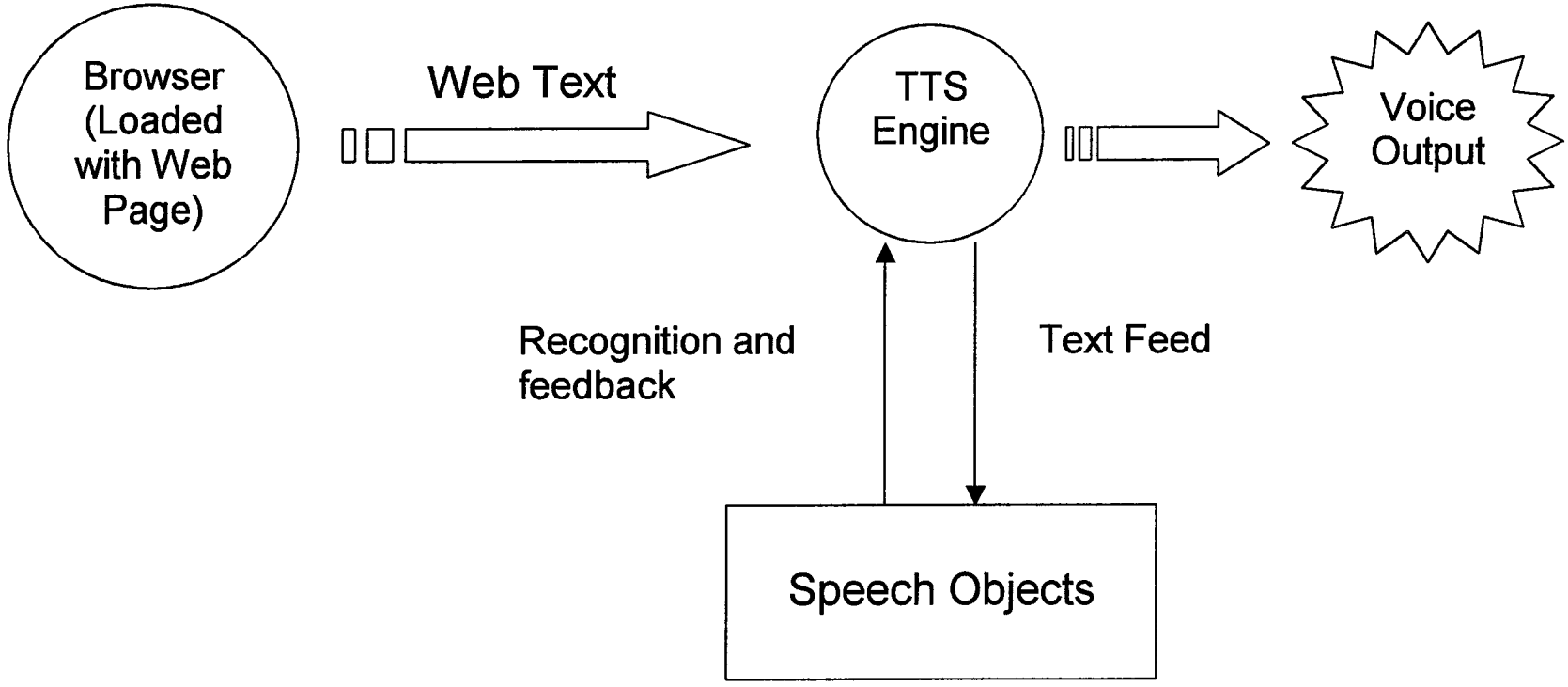
Full System Architectural Diagram:



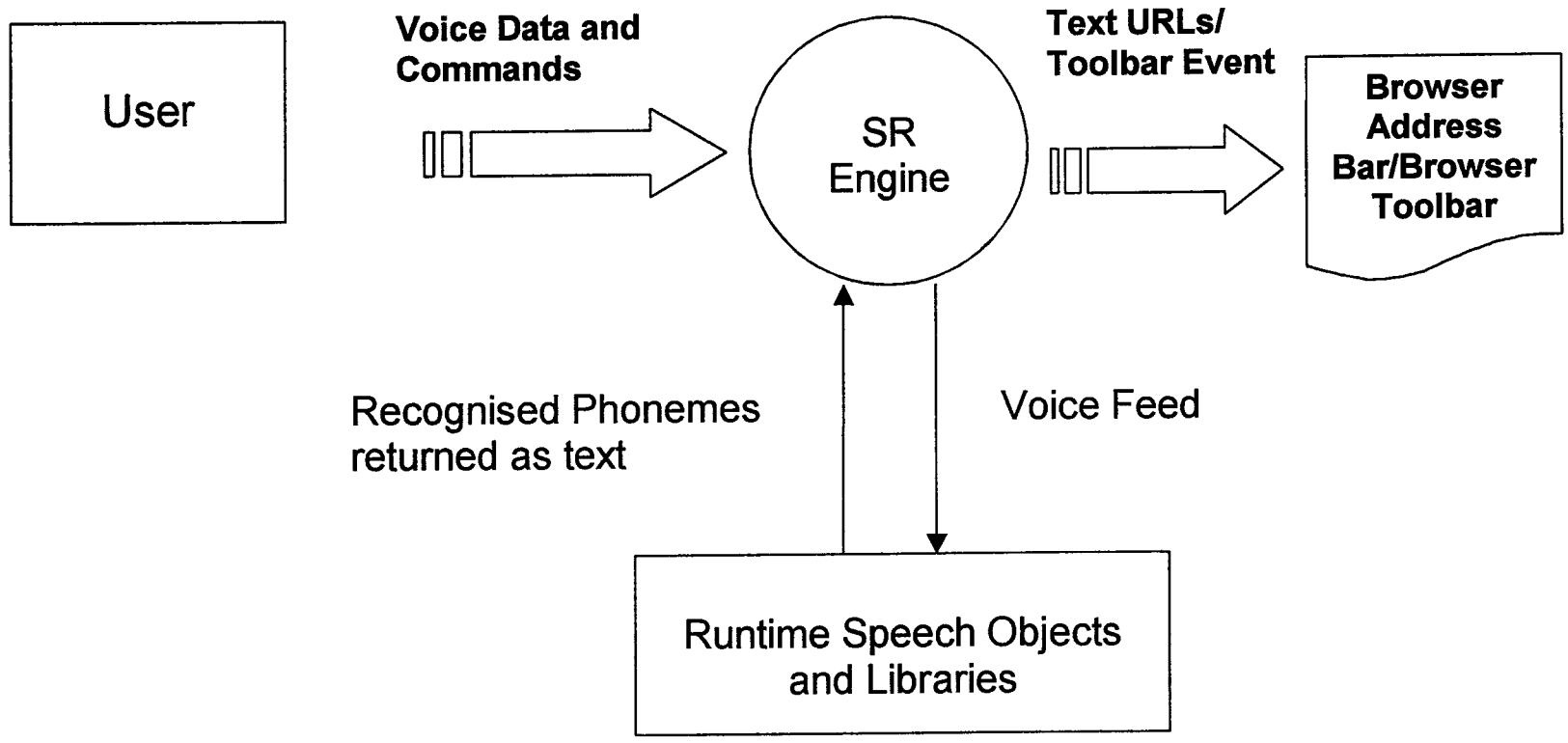
Standalone Browser Unit Process Diagram:



Text-To-Speech Engine:



Speech Recognition Engine:



Module Level Concepts

Module Level Concepts:

The Speech Enabled Browser follows a Prototype model. Although the programming language (Visual Basic) used in this project is effective by itself for prototyping, the project was designed by developing several small modules or prototypes and tested before extensions were made.

There are 3 modules in the system:

- The Browser Module
- The Text-To-Speech unit
- The Speech Recognition unit

Modularity in a System is defined by 2 main factors:
Coupling and Cohesiveness.
If these factors can be scaled into LOW, MEDIUM and HIGH levels, then the modules can be studied as follows:

- The Browser Module

  Coupling Level : Low, Not dependent on other units and can function both as a standalone browser.

  Cohesive Level : Medium, performs Web Page Navigation, File Operations, etc.

- The TTS (Text-To-Speech) Module

  Coupling Level : High, depends heavily on the GUI Browser Interface for it's functioning.

  Cohesive Level : Low, performs the single unique function of vocally reading a Web Page's Text.

- The Speech Recognition Module

  Coupling Level : High, also depends on the GUI Browser Interface for it's proper working.

  Cohesive Level : Medium, performs the twin task of detecting voices and converting it to text.

It maybe concluded from the above analysis that the most necessary component i.e. the Browser is independent of the other components and hence adds a lot of stability to the system.

In the unlikely event of the other 2 modules failing, the Browser unit will still function properly.

QUALITY ASSURANCE

Quality Assurance:

Being a utility software the speech enabled browser has been strictly put through testing processes that vouchsafe its effective usage.

Unit testing:

The unit testing phase of this software was divided into 3 blocks, one for each module.
Prototypes of the modules were made and they were each tested for various parameters:

Browser Module: tested for efficient page loading, speed, Graphical Interface support, size and ease of portability.

Text-To-Speech module: tested independently for voice clarity, speed, efficiency of system resource usage (especially media controls), ambiguity in speech, pronunciation and interface support.

Speech Recognition module: tested for speed of recognition vs. the accuracy of phrases, loading time, pattern matching, detection and filtering of interference and interface support.

Integration Testing:

The 3 modules were combined to form the software as a whole and this combination was again tested for general efficiency.

- The parameters for this testing were:
- Size and Portability
- Dependency on other components
- Efficiency on an actual network environment (Internet)
- Ease of using speech systems during runtime
- Error handling and web page usage.

MERITS AND DEMERITS

Merits and Demerits:

Using the Speech Enabled Browser provides users with a whole new array of advantages. Some of these may exhibit simplicity but the effectiveness and utility value of the software cannot be doubted.

For the home PC:

- Simple browsing capability with less complexity of usage.
- Easy-to-use Graphical User Interface.
- Speech makes it an educative and informative experience.
- Fancy and novel utility software.

For commercial computers:

- Although less powerful compared to other browsers, it has an edge with its Speech system built in.
- Small and simple utility that can be easily ported with most modern systems.
- Speech concepts with its technological advancement make its support and dependent components, *Future-Friendly*.
- Using speech recognition increases the degree of user friendliness between the user and the network environment.
- The real benefit is for Visually Impaired users whose lack of vision often prevents them from using the vast resources of the online world.
- The usage of speech components provides an opportunity to access web space in a novel and technologically improved manner.

General Merits:

- The Speech Enabled Browser will be distributed as freeware software, enhancing its usability.
- Use of simple and effective coding methods makes room for further improvements of the software.

Unfortunately, like most application software, the Speech Enabled Browser too possesses certain demerits that exist more as Trade-Off's rather than lack of solutions.

- The newness of Speech Technology makes it, sometime, difficult to work with.
- Speech systems require fast computers to run on.
- The everlasting conflict between Accuracy and Speed of Recognition will continue to exist.
- The software depends on its speech libraries and Engines that must be shipped along with it. (Although this is not a problem as such, it increases the dependency of the software).
- Compatibility with high grade, fast systems that exhibit good Multimedia capabilities; little or no backward compatibility.
- Lesser speed and power compared to other browsers like Opera, Internet Explorer, Netscape, etc.
- Being in the nascent stage of development, fault tolerance, pattern matching and user voice profiles may not be high.

ENHANCEMENT PLAN

Enhancement plan:

Although this software with its "future-friendliness" is quite safe for the years to come, it must have enhancements created periodically to keep its place as utility software high and running.

Some of the proposed enhancements for this software are:

- Faster recognition of speech (this can come only with improved speech engines)
- Complete control of the browser via speech (for now, only two aspects: speaking addresses and reading the page have been covered).
  The future plan includes control of menus, buttons and any object on the interface via speech.
- Automatic "Voice-Login" for e-mail accounts. Although this feature would breach security, it adds to the general usability of the software.
- Voice feedback, currently covers only static pages and simple web text. In the future, it may cover dynamic pages, animated text and even captions on screen objects.
- Greater interface, especially use of "Skins" would enhance viewing and usage of the software.
- Database capability with URLs dynamically accessed and used along with the speech.
- Automatic e-mail alerts and e-mail specific customization that will help users with their mail accounts.

Hopefully this enhancement plan will make the Speech Enabled Browser a better and more effective utility.

Conclusion

Conclusion

It has been a great experience working on this project. From the beginning the excitement has always been there, the concept being relatively unknown, making it all the more challenging.

The Speech Enabled Browser may not live up to the expectations made from being a utility but it certainly will make a place for itself with its innovative and value added nature.
The very idea behind the development of this software was to provide users with a tool that could be used for some constructive purpose and at the same time, to use the latest technology so as to make it obsolescence-proof.

The speech enabled browser not only connects the user with the internet but it does so with a set of fancy works in the form of speech. It has gone a step closer to imitating human emotions and bringing users closer to their own nature in a machine.

This project is the combination of academic study, innovative ideas and the need to implement knowledge gained. Hopefully, the Speech Enabled Browser will reflect these qualities by being a useful tool for all, for the years to come.

BIBLIOGRAPHY

Bibliography:

The following reference texts were useful in developing this software:

Learn Visual Basic 6.0 Now - by Michael Halvorson

The SAPI help texts - by Microsoft Corporation

An integrated approach to Software Engineering - by Pankaj Jalote

Online Documents from VisualBasicForum.com and VB-Helper.com

Software used:

The important design and development software tools used in the creation of this project are:

Design and Integration: Microsoft Visual Basic 6.0 by Microsoft Corporation

Internal Components : Microsoft Speech SDK version 5.1 By Microsoft corporation

Enhancement tool (Icon creators) : "AW Icons" icon editor by Lokas Ltd.

Help file utilities : RoboHelp by Blue Sky software Corp.

Appendix 1: Sample Code

Sample Code:

```
Option Explicit
  Dim arradd(10) As String
  Dim c As Integer
  Dim TState As Boolean
  Dim voice As SpVoice
  Dim V As SpeechLib.SpVoice
  Dim i As Integer
  Dim token As ISpeechObjectToken

Private Sub cmdExit_Click()
 End
End Sub


Private Sub AdvVol_Click()
 RunUI SpeechAudioVolume
End Sub

Public Function VoiceChange()
 Set voice.voice = voice.GetVoices().Item(Voice3.Index)
 voice.Speak "The voice has been changed to " & Voice3.Tag
End Function


Private Sub Form_Load()

 c = -1
 SetState False
 Set V = New SpeechLib.SpVoice
 Set voice = New SpVoice

 voice.Speak "Welcome, to the Speech Enabled Browser!!",
 SVSFlagsAsync
 VolSlide.Min = 0
 VolSlide.Max = 100
 VolSlide.Value = V.volume

 RateSlide.Min = -10
 RateSlide.Max = 10
 RateSlide.Value = V.Rate

 SilenceCmd.Enabled = True
 Open "c:\windows\desktop\History.txt" For Output As #1
```

```
 Addbar1.AddItem "c:\my documents\symbi.htm"
 Addbar1.AddItem "http://www.microsoft.com"
 Addbar1.AddItem "http://www.rediffmail.com"
 Addbar1.AddItem "http://www.yahoomail.com"
 Addbar1.AddItem "http://www.google.com"
 WebBrowser1.Navigate ("about:blank")

End Sub

Private Sub BackCmd_Click()
 On Error GoTo ErrHandler:
  If c > 0 Then
        c = c - 1
       WebBrowser1.Navigate arradd(c)
        Addbar1.Text = arradd(c)
 End If
  ForwardCmd.Enabled = True
  Exit Sub
ErrHandler:
 BackCmd.Enabled = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
 Unload Form2
 Unload Form3
 Unload Form4
 Unload Form5
 Unload Form6
End Sub


Private Sub ForwardCmd_Click()
  On Error GoTo ErrHandler:
  If c < 9 Then
        c = c + 1
       WebBrowser1.Navigate arradd(c)
        Addbar1.Text = arradd(c)
 End If
  BackCmd.Enabled = True
  Exit Sub
ErrHandler:
 ForwardCmd.Enabled = False
End Sub
```

```
Public Sub GoCmd_Click()
 WebBrowser1.Navigate Addbar1.Text
 Write #1, WebBrowser1.LocationURL
 If c < 9 Then
    c = c + 1
    arradd(c) = Addbar1.Text
 End If
  BackCmd.Enabled = True
  StatusBar1.Panels(1) = "URL: " + Addbar1.Text + "    "
End Sub


Private Sub HomeCmd_Click()
 WebBrowser1.Navigate ("about:blank")
 BackCmd.Enabled = True
 Addbar1.Text = "about:blank"
 c = c + 1
End Sub


Private Sub mnuAbout_Click()
 Form3.Show
End Sub


Private Sub mnuBack_Click()
 Call BackCmd_Click
End Sub


Private Sub mnuCopy_Click()
 On Error Resume Next
 WebBrowser1.SetFocus
 WebBrowser1.ExecWB OLECMDID_COPY, OLECMDEXECOPT_DODEFAULT
End Sub


Private Sub mnuCut_Click()
 On Error Resume Next
 WebBrowser1.SetFocus
 WebBrowser1.ExecWB OLECMDID_CUT, OLECMDEXECOPT_DODEFAULT
End Sub


Private Sub mnuExit_Click()
 End
End Sub
```

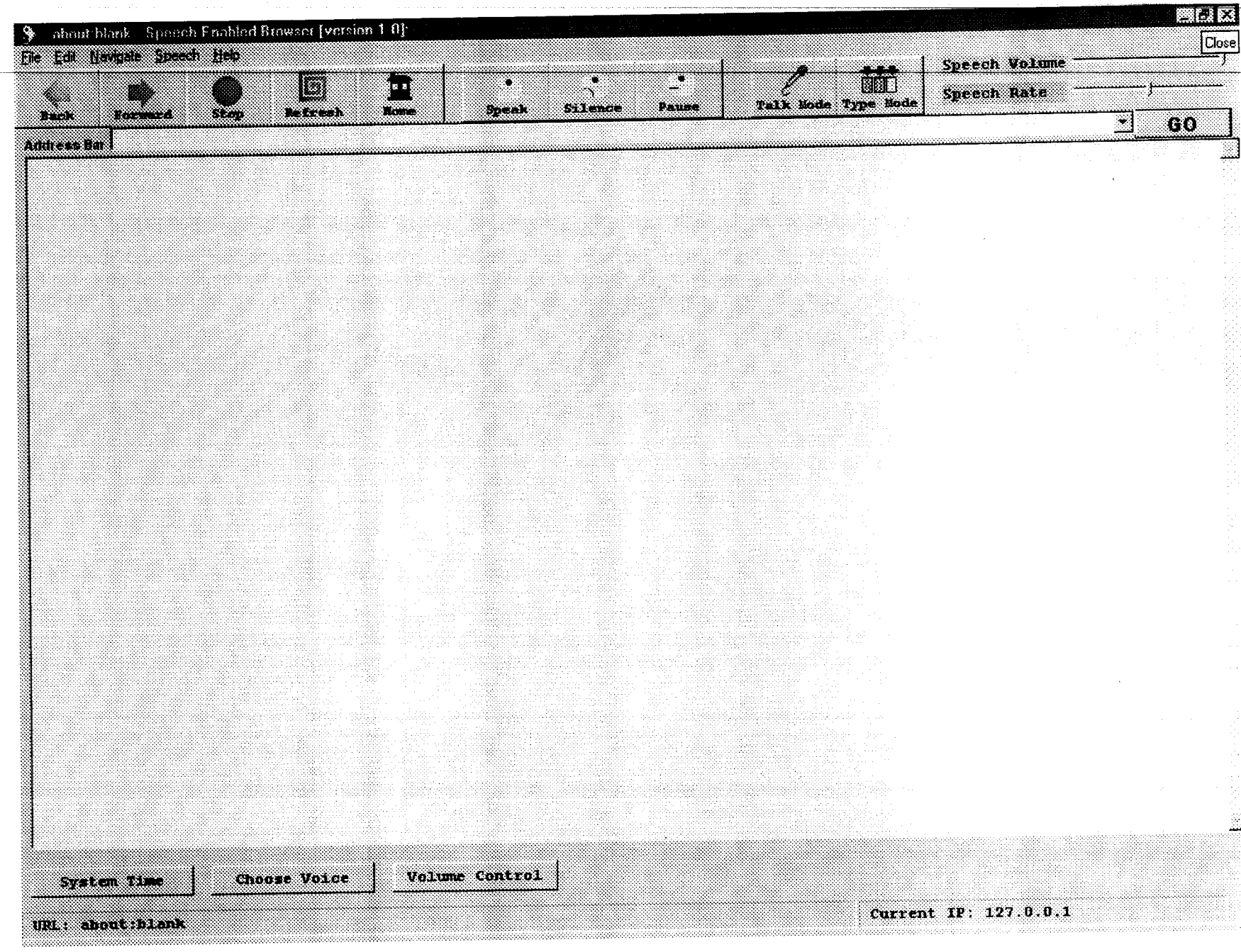
```
Private Sub mnuForward_Click()
 Call ForwardCmd_Click
End Sub


Private Sub mnuGo_Click()
 Call GoCmd_Click
End Sub


Private Sub mnuHome_Click()
 Call HomeCmd_Click
End Sub


Private Sub mnuNew_Click()
 Dim web As Form1
 Set web = New Form1
 web.Show
End Sub


Private Sub mnuOpen_Click()
 cd1.CancelError = True
 On Error GoTo ErrHandler:
 cd1.DialogTitle = "File Open"
 cd1.Filter = "HTML Pages (.html)|*.html|" & _
              "HTML Pages (.htm)|*.htm|" & _
              "GIF Images (.gif)|*.gif|" & _
              "Text Files (.txt)|*.txt|" & _
              "All Files|*.*"
 cd1.ShowOpen
 WebBrowser1.Navigate (cd1.FileName)
ErrHandler:
End Sub


Private Sub mnuPaste_Click()
 On Error Resume Next
 WebBrowser1.SetFocus
 WebBrowser1.ExecWB OLECMDID_PASTE, OLECMDEXECOPT_DODEFAULT
End Sub
```

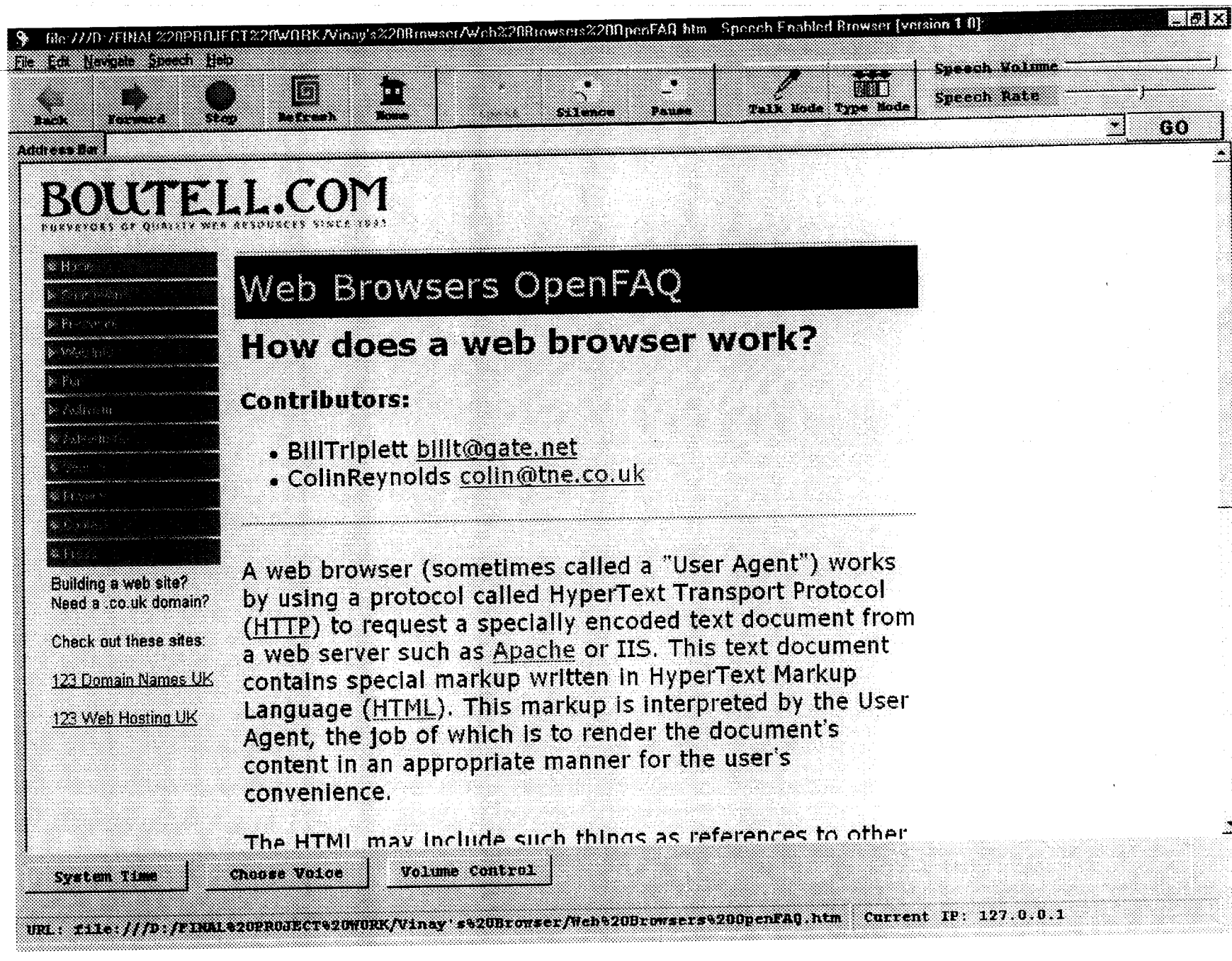
```
Private Sub mnuPause_Click()
 Call PauseRes_Click
End Sub


Private Sub mnuRefresh_Click()
 Call RefreshCmd_Click
End Sub


Private Sub mnuResume_Click()
 Call PauseRes_Click
End Sub


Private Sub mnuSave_Click()
 WebBrowser1.SetFocus
 On Error Resume Next
 WebBrowser1.ExecWB OLECMDID_SAVEAS, OLECMDEXECOPT_DODEFAULT
End Sub


Private Sub mnuSelectAll_Click()
 On Error Resume Next
 WebBrowser1.SetFocus
 WebBrowser1.ExecWB OLECMDID_SELECTALL, OLECMDEXECOPT_DODEFAULT
End Sub


Private Sub mnuSpeakRead_Click()
 Call SpeakCmd_Click
End Sub


Private Sub mnuStop_Click()
 Call StopCmd_Click
End Sub


Private Sub mnuStopRead_Click()
 Call SilenceCmd_Click
End Sub


Private Sub mnuTalkMode_Click()
 Call TalkCmd_Click
End Sub
```

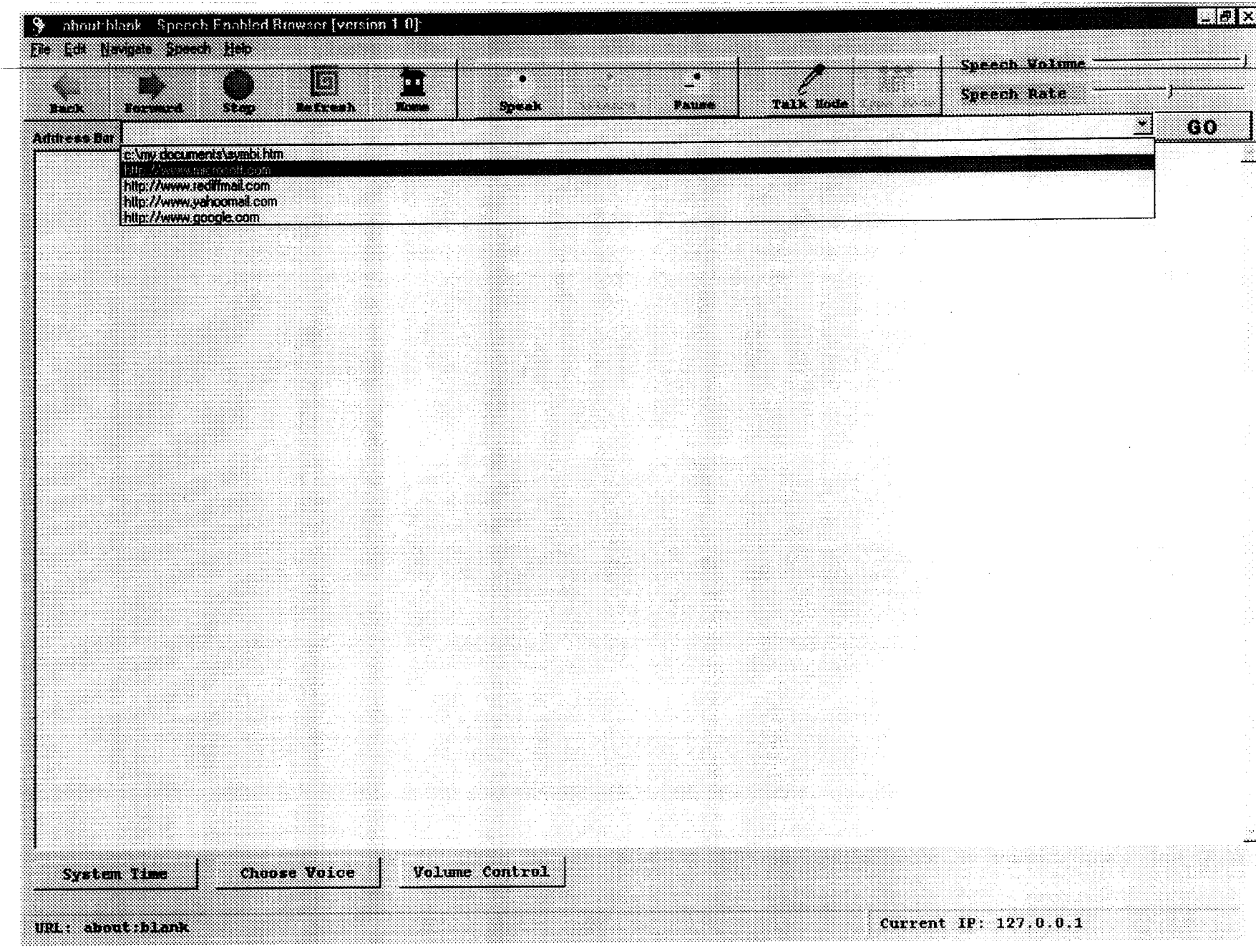
```
Private Sub mnuTypeMode_Click()
 Call TypeCmd_Click
End Sub


Private Sub RefreshCmd_Click()
 WebBrowser1.Refresh
End Sub


Private Sub PauseRes_Click()
 If PauseRes.Caption = "Pause" Then
  PauseRes.Caption = "Resume"
  mnuPause.Enabled = False
  mnuResume.Enabled = True
  V.Pause
 Else
  PauseRes.Caption = "Pause"
  mnuResume.Enabled = False
  mnuPause.Enabled = True
  V.Resume
 End If
End Sub


Private Sub SilenceCmd_Click()
  V.Speak vbNullString, SVSFPurgeBeforeSpeak
  SilenceCmd.Enabled = False
  mnuStopRead.Enabled = False
  SpeakCmd.Enabled = True
  mnuSpeakRead.Enabled = True
End Sub


Private Sub SpeakCmd_Click()
 Clipboard.Clear
 WebBrowser1.SetFocus
 WebBrowser1.ExecWB OLECMDID_SELECTALL, OLECMDEXECOPT_DODEFAULT
 WebBrowser1.ExecWB OLECMDID_COPY, OLECMDEXECOPT_PROMPTUSER
 WebBrowser1.ExecWB OLECMDID_CLEARSELECTION,
OLECMDEXECOPT_DODEFAULT
 Form2.WebText = Clipboard.GetText

 On Error Resume Next
  V.Speak Form2.WebText.Text, SVSFlagsAsync
  SpeakCmd.Enabled = False
  mnuSpeakRead.Enabled = False
  SilenceCmd.Enabled = True
  mnuStopRead.Enabled = True
End Sub
```

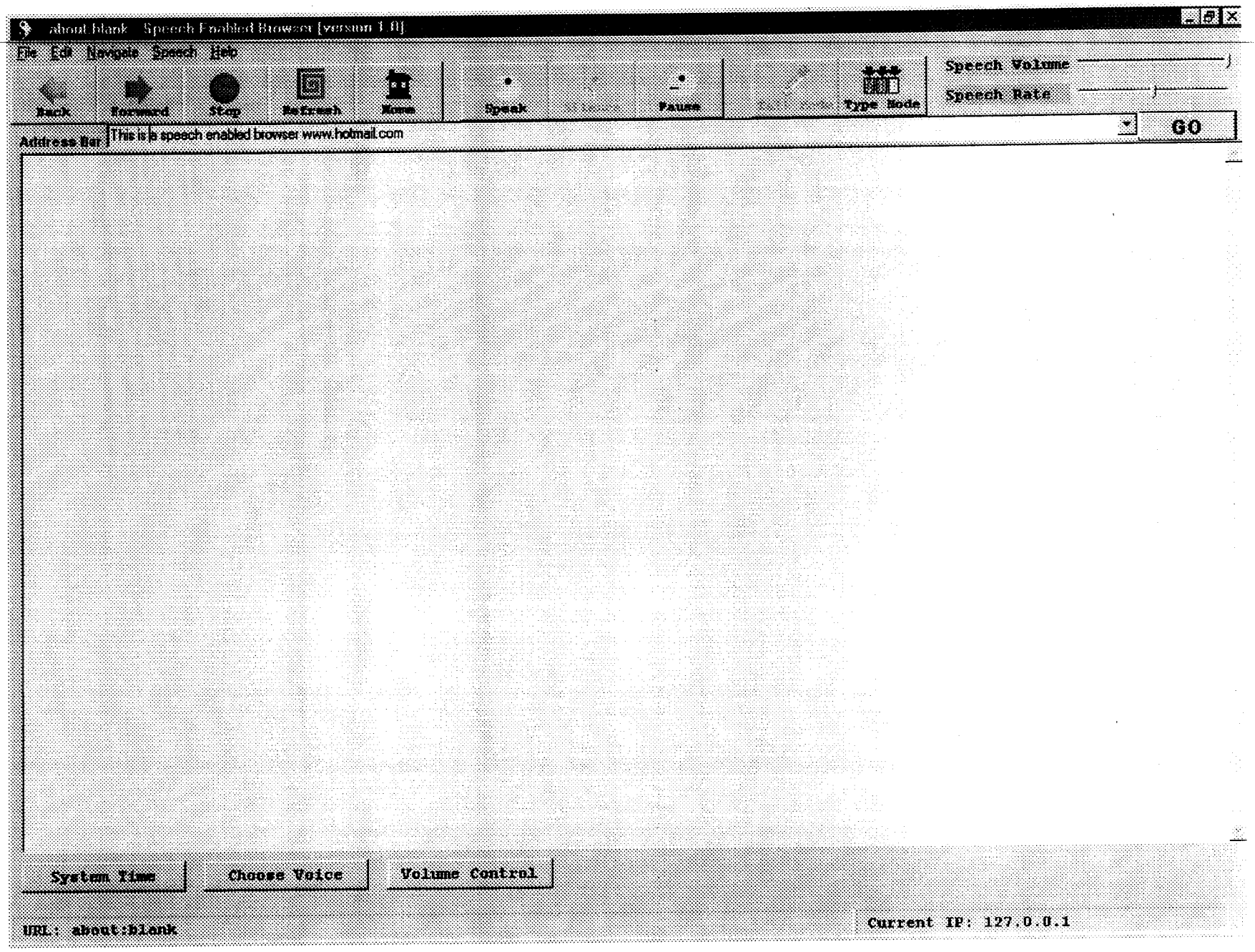
```
Private Sub SetState(ByVal NewState As Boolean)
    TState = NewState
    SpeakCmd.Enabled = Not TState
    mnuSpeakRead.Enabled = Not TState
    SilenceCmd.Enabled = TState
    mnuStopRead.Enabled = TState
End Sub


Private Sub StopCmd_Click()
 WebBrowser1.Stop
End Sub


Private Sub mnuPrint_Click()
 WebBrowser1.SetFocus
 On Error Resume Next
 WebBrowser1.ExecWB OLECMDID_PRINT, OLECMDEXECOPT_PROMPTUSER
End Sub


Private Sub SysTime_Click()
 voice.Speak "The Current time is " & Time, SVSFlagsAsync
End Sub


Private Sub TalkCmd_Click()
 Call Form4.Talk
 Form4.SetTState True
End Sub


Private Sub TypeCmd_Click()
 Call Form4.TypeMode
 Form4.SetTState False
End Sub


Private Sub VoiceSwitch_Click()
 Load Form6
 Form6.Visible = True
End Sub
```

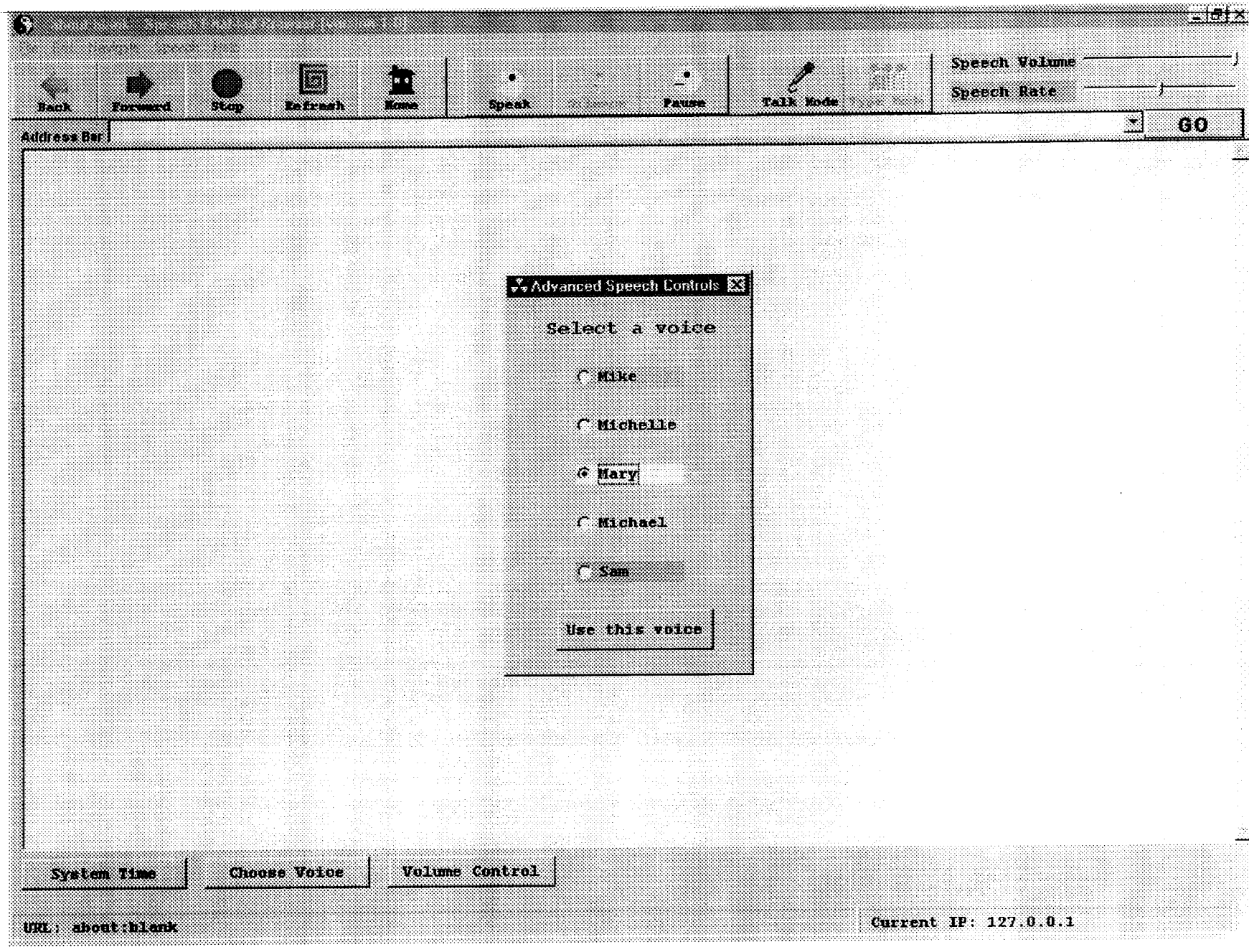
Appendix 2: Screenshots


about:blank - Speech Enabled Browser [version 1.0]
Close
File Edit Navigate Speech Help
Back
Forward
Stop
Refresh
Home
Speak
Silence
Pause
Talk Mode
Type Mode
Speech Volume
Speech Rate
GO
Address Bar
System Time
Choose Voice
Volume Control
URL: about:blank
Current IP: 127.0.0.1


file:///D:/FINAL%20PROJECT%20WORK/Vinay's%20Browser/Web%20Browsers%20OpenFAQ.htm Speech Enabled Browser [version 1.0]
File Edit Navigate Speech Help
Back
Forward
Stop
Refresh
Home
Silence
Pause
Talk Mode
Type Mode
Speech Volume
Speech Rate
GO
Address Bar
BOUTELL.COM
Building a web site?
Need a .co.uk domain?
Check out these sites:
123 Domain Names UK
123 Web Hosting UK
Web Browsers OpenFAQ
How does a web browser work?
Contributors:
• BillTriplett billt@gate.net
• ColinReynolds colin@tne.co.uk
A web browser (sometimes called a "User Agent") works by using a protocol called HyperText Transport Protocol (HTTP) to request a specially encoded text document from a web server such as Apache or IIS. This text document contains special markup written in HyperText Markup Language (HTML). This markup is interpreted by the User Agent, the job of which is to render the document's content in an appropriate manner for the user's convenience.
The HTML may include such things as references to other
System Time
Choose Voice
Volume Control
URL: file:///D:/FINAL%20PROJECT%20WORK/Vinay's%20Browser/Web%20Browsers%20OpenFAQ.htm Current IP: 127.0.0.1


about:blank - Speech Enabled Browser [version 1.0]
File Edit Navigate Speech Help
Back
Forward
Stop
Refresh
Home
Speak
Pause
Talk Mode
Speech Volume
Speech Rate
Address Bar
GO
c:\my documents\srmbi.htm
http://www.rediffmail.com
http://www.yahoomail.com
http://www.google.com
System Time
Choose Voice
Volume Control
URL: about:blank
Current IP: 127.0.0.1


about:blank - Speech Enabled Browser [version 1.0]
File Edit Navigate Speech Help
Back
Forward
Stop
Refresh
Home
Speak
Pause
Type Mode
Speech Volume
Speech Rate
Address Bar
This is a speech enabled browser www.hotmail.com
GO
System Time
Choose Voice
Volume Control
URL: about:blank
Current IP: 127.0.0.1


Back
Forward
Stop
Refresh
Home
Speak
Pause
Talk Mode
Speech Volume
Speech Rate
GO
Address Bar
Advanced Speech Controls
Select a voice
Mike
Michelle
Mary
Michael
Sam
Use this voice
System Time
Choose Voice
Volume Control
URL: about:blank
Current IP: 127.0.0.1