

WEB-BASED EDUCATION:  
A SPEECH RECOGNITION AND SYNTHESIS TOOL

by

LAURA SCHINDLER

Advisor  
DR. HALA ELAARAG

A senior research proposal submitted in partial fulfillment of the requirements  
for the degree of Bachelor of Science  
in the Department of Mathematics and Computer Science  
in the College of Arts and Science  
at Stetson University  
DeLand, Florida

Fall Term  
2005

## **ACKNOWLEDGMENTS**

There are many people who I wish to acknowledge and thank for their timeless support and help. First, thank you to Dr. ElAarag for guiding me throughout the process and providing thoughtful insight. Thank you also to all of the professors who spoke with me while I tried to decide my topic. Next, thank you to my friends, who have remained on amicable terms with me despite my long hours and (to them) gibberish spiels about speech patterns and coding problems. I would like to especially thank my roommate, who had to tolerate my incessant typing in the wee hours of the morning, my cursing, and my cries of annoyance and (eventually) happiness. Finally, thank you to my family for supporting me in every manner possible. Without them I would have been unable to do so much.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	ii
TABLE OF CONTENTS.....	iii
LIST OF FIGURES .....	iv
ABSTRACT.....	1
1. INTRODUCTION .....	2
2. Background.....	4
2.1 Speech Synthesis.....	4
2.2 Speech Recognition .....	4
3. Related Work .....	6
4. Implementation .....	8
4.1 GUI .....	8
4.2 Networking .....	8
4.3 Speech Synthesis.....	9
4.4 Speech Recognition .....	9
5. Conclusion .....	11
6. Future Work.....	12
APPENDIX A: SPEECH GLOSSARY .....	13
APPENDIX B: APPLICATION SCREENSHOTS.....	14
B.1 Disconnected User Images.....	14
B.2 Connected User Images.....	16
B.3 Server Images.....	18
REFERENCES .....	19

## LIST OF FIGURES

Figure B.1: Disconnected User Window .....	14
Figure B.2: Opening a file .....	15
Figure B.3: Saving text to a file .....	15
Figure B.4: Connection Preferences .....	16
Figure B.5: Failed to connect because of IP .....	16
Figure B.6: Connected Client .....	17
Figure B.7: Initialized server .....	18

## **ABSTRACT**

Many of the new technologies designed to help worldwide communication – e.g. telephones, fax machines, computers – have created new problems especially among the hearing and visually impaired. A person, who has severe hearing impairments, particularly to the extent in which deafness occurs, may experience difficulties communicating over a telephone as he or she is unable to hear the recipient's responses. Conversely, someone with visual impairments would have little inconvenience using a telephone but may not be able to communicate through a computer because of the difficulties (or, in the case of blindness, impossibility) in reading the screen. The goal of this project is to incorporate current speech recognition (speech-to-text) and speech synthesis (text-to-speech) technology into a chat room, thus, providing a solution to communication between the hearing and visually impaired that is free and does not require any additional equipment besides a computer. Additionally, it is hoped that this application may also be used in educational settings, regardless of students' or teachers' disabilities, as a teaching aid.

## 1. INTRODUCTION

For most people, communication with others is quite simple. There are many options available: telephones, mail, electronic mail, chat rooms, instant messaging, etc. However, this becomes a more difficult task for those with disabilities. A deaf person does not have the luxury of being able to dial someone on the telephone and talk without additional equipment. Similarly, it is difficult for a blind person to communicate through mail or electronic means that require being able to see the screen. Direct communication between a deaf and blind person is almost impossible without a mediator.

There currently does not appear to be much software that directly addresses communication between the blind and the deaf. Much of the technology found is quite expensive, involves additional hardware, or uses Braille rather than speech. For example, Freedom Scientific offers a package solution, which allows for communication face-to-face, over telephones, and over the Internet; however, this costs \$6,705 [1]. The goal of this project is to incorporate current speech recognition (speech-to-text) and speech synthesis (text-to-speech) technology into a chat room, thus, providing a solution to communication between the deaf and blind that is both free and does not require any additional equipment besides a computer.

There were several programming languages that could have been used in the creation of this project. Ultimately, C++ was chosen because of its speed, cross-systems capabilities, and the fact that well-tested packages pertaining to speech recognition and synthesis were found to be written in C++ and C. Due to time constraints, it would have been unwise to create a speech synthesis or recognition program from scratch. A search of the Internet produced several packages, which had been written over the course of several years and involved groups of highly skilled individuals who specialized in speech recognition and synthesis. Two of the packages found, Festival [2], and Sphinx-3 [3] were incorporated into the project. The Festival Speech Synthesis System, written in C++, offers a framework for building a speech synthesis system. Sphinx-3 provides the means for the speech recognition aspects. It is written in C so minor adjustments will need to be made for it to work well with the, otherwise, C++ code.

There are several aspects of the application that must be addressed in addition to the speech synthesis and recognition. First, it will involve a client and server with a graphical user interface (GUI) for the client. SDL\_net was chosen to handle the networking while Gtkmm was used to create the GUI. Secondly, additional planned features include multiple voices so that individuals do not all sound the same, thus allowing easier recognition (at a minimum, one male voice and one female voice); easy customization to allow for phonetic pronunciations to be associated with new words or names; recognition of basic chat lingo (i.e. emoticons, lol, rotfl, etc.); and the ability to load pre-existing (or custom-made) voices. Lastly, the conversations will also be savable for future review or use.

In addition to allowing communication between the blind and deaf, this project has many other applications. For example, a teacher may wear a headset with a microphone while lecturing. The lecture could then be saved as a text file that any student could access during the lecture (if they are linked through a computer) or afterwards. Additionally, the speech synthesis aspect of the application could be used as a tool for learning phonics.

In order to ensure that this software is developed in such a manner that it may be useable by those with disabilities, the researcher has opened communication with Karen Cole, the director of Academic Resources at Stetson University. Dr. Cole has offered her assistance as well as the assistance of current Stetson students with impaired vision or hearing to be used as a future resource for testing and development of the software to ensure that the needs of the disabled are adequately met.

This paper contains five main sections: background information, related work, implementation, conclusion, and future work. The background section provides a general knowledge of what speech synthesis and recognition entails. Some of the past works concerning web-based education and applications of current speech synthesis and recognition are described in the following section. The third section describes the various packages and techniques used in the implementation of the application created in this research. Finally, the conclusion discusses the results that have thus far been attained in the creation of an application while the future works section describes what will be implemented in spring semester.

## **2. Background**

The main focus of this application is to provide an environment in which speech synthesis and recognition may be successfully implemented. This is done in the hopes that the accuracy ratings of this system would range in the 90-percentile. In order to accomplish, it was essential to develop strong background knowledge of the current processes and terminologies associated with speech synthesis and recognition.

### **2.1 Speech Synthesis**

As Text-to-Speech implies, speech synthesis involves two basic processes: the reading in of text and the production into sound. For simplification purposes, call these the front end and back end, respectively. First, the front end must read the text and transform any numbers or abbreviations into text. For example, “lol” would be changed to “laugh out loud.” A phonetic transcription is then ascribed to each word using text-to-phoneme (TTP) or grapheme-to-phoneme (GTP) processes [6]. How a text should be spoken, including the pitch, frequency, and length of the phonemes is determined in this stage [7]. This makes up the symbolic linguistic representation. The back end then takes that representation and attempts to convert it into actual sound output according to the rules created in the front end.

Speech synthesis has little, if any, understanding of the actual text being read. Such software is, typically, not concerned with what a sentence or word actually means. Rather, it simply uses dictionaries or rules to make guesses as to how the text should be read [7]. Text-to-phoneme conversion guesses the pronunciation by using either the dictionary-based approach or the rule-based approach. In the dictionary-based approach, a large dictionary of words and spellings is stored by the program and accessed at appropriate times. This method, however, is very space consuming. The other option, rule-based approach uses preset rules of pronunciation to sound out how a word should be pronounced. Most speech synthesizers use a combination of both approaches [6].

While the previously mentioned methods certainly help the computer, it is difficult to determine pronunciation without grasping the meaning. For example, how should the front end translate 1983? If it is used in the sentence, “There are 1983 students,” then it would be pronounced “one thousand and eighty-three.” However, if it was in the sentence, “She was born in 1983,” it is pronounced “nineteen eighty-three.” It is almost impossible for a computer to decipher pronunciations, especially when both pronunciations are used in the same sentence. (i.e. In 1983, 1983 ducks swam the English Channel.) Similar problems exist for words which have two pronunciations, such as read (pronounced rēd or rĕd). This has yet to be perfected, and errors are still common [6].

### **2.2 Speech Recognition**

Speech recognition allows a computer to interpret any sound input (through either a microphone or audio file) to be transcribed or used to interact with the computer. A speech recognition application may be used by a large amount of users without any

training or may be specifically designed to be used by one user. In this speaker-dependent model, accuracy rates are typically at their highest with approximately a 98% rate (that is, getting two words in a hundred wrong) when operated under optimal conditions (i.e. a quiet room, high quality microphone, etc) [8].

Generally, modern speech recognition systems are based on the hidden Markov models (HMMS). HMMS is a statistical model which attempts to determine the hidden components by using the known parameters. For example, if a person stated that he wore a raincoat yesterday, then one would predict that it must have been raining. Using this technique, a speech recognition system may determine the probability of a sequence of acoustic data given one word (or word sequence). Then, the most likely word sequence may be determined using Baye's rule:

$$\Pr(\text{word} \mid \text{acoustics}) = \frac{\Pr(\text{acoustics} \mid \text{word}) \Pr(\text{word})}{\Pr(\text{acoustics})}.$$

According to this rule, for any given sequence of acoustic data (for example, an audio file or microphone input),  $\Pr(\text{acoustics})$  is a constant and, thus, ignorable.  $\Pr(\text{word})$  is the prior probability of the word according to a language modeling. [As an example, this should ensure that  $\Pr(\text{mushroom soup}) > \Pr(\text{much rooms hope})$ .]  $\Pr(\text{acoustics} \mid \text{word})$  is obtained using the aforementioned HMMS [8].

While recognition of words has risen to 80-90% (depending on the location), grammar remains less focused upon and, thus, less accurate. In order to determine punctuation, it is necessary to differentiate between the stressed syllables or words in an utterance. For instance, when naturally speaking, it is easy to differentiate between "Go." "Go!" and "Go?" However, most speech recognition systems solely provide what word was uttered and do not note stress or intonation. As such, this information cannot be used by the recognizer. The most frequent solution is to simply require a user to announce when and where punctuation should occur.

### 3. Related Work

In the past decade there has been an increasing amount of work dedicated to web-based education. Benefits of such systems are clear: distance is no longer an issue, feedback is expedient, and assignments may be catered to specific classes or individuals. Applications such as Blackboard system have been implemented in many universities to work in conjunction with classes. However, the focus has shifted from such static environments to more adaptive ones, which would alter teachings or assign different homework according to the assessed level of the individual students.

Several systems have been proposed to change web-based education from meaning a reference of hyperlinks to an interactive system involving adaptation and artificial intelligence. Generally, there are two types of adaptive educational systems that are employed: intelligent tutoring systems (ITS) and adaptive hypermedia systems. ITS technologies involve curriculum sequencing, intelligent analysis of student's solutions, and interactive problem solving support. The goal of the system is to provide an optimal path for a student to reach a goal lesson using a series of questions, problems, and examples. These are presented in varying orders and difficulties according to what the student is accurately able to answer. If an answer is incorrect, the system should be able to figure out where the student went wrong and provide detail information about the mistakes followed by additional examples and lessons. Adaptive hypermedia systems simply adjust what links are shown according to the student's learning level. [9]

Additional web-based education systems include virtual environments, which provide a multimedia experience without the student leaving the computer, and whiteboards. Virtual environments can include field trips through museums or historical sites or scientific experiments and dissections. Interactivity allows the student to observe at his or her own speed and learn more about specific areas at a click of the mouse. Whiteboards provide a space for the user to type, draw, or present other data that can be viewed by anyone else connected to the board. In this manner, the student is not limited to simple text communication but can easily include his or her own drawings and images. Speech synthesis and recognition may also be used in web-based education to provide another means for communication and interactivity.

It is necessary to differentiate between speech synthesis and speech recognition. While it is true that both involve a computer and its interpretation of human speech, the reversal of input and output between recognition and synthesis differs enough that much work focuses on one, rather than simultaneously develop tools for both (i.e. Festival, Sphinx, ModelTalker[10], Ventrilo[11]...). This is exemplified by the fact that two different packages were used in the speech synthesis and recognition aspects of this project. However, a connection does exist between the two. Mari Ostendorf and Ivan Bulyko discuss how the two are intertwined and may influence the advancement of the other now and in the future. [12]

The emphasis on speech recognition has shifted over the past decade from a tool to study linguistic development to emphasizing engineering techniques. Today, the concern focuses more on optimization: getting the most accurate results in the fastest amount of time. This shift has produced considerable advancement although error rates are still fairly high. For instance, the word error rates on broadcast news are 13% while conversational speech and meeting speech produces error rates of 24% and 36% respectively. These rates further increase under noisy conditions. [13]

Various techniques are implemented to increase the accuracy of speech recognition including mel-cepstral speech analysis, hidden Markov modeling (HMMS), clustering techniques, n-gram models of word sequences, a beam search to choose between candidate hypotheses, acoustic models which adapt to better match a single target speaker, and multi-pass search techniques to incorporate adaptation as well as models of increased complexity.[9] Speech synthesis has adopted the search algorithms found in speech recognition. Rather than relying on a sole instance of each unit in the database, speech synthesis now often incorporates multiple instances to allow for more choices and increase the quality of the concatenate synthesizer. Selecting the unit is implemented using the Viterbi search, which is, somewhat, a reversal of the decoding process in speech recognition. Instead of finding the word or word sequence that most closely matches the audio input, unit selection search finds a sequence of acoustic input that optimally matches a given word or word sequence. Both techniques involve concatenating context-dependent phonetic subword unit into words although synthesis must also include proper lexical stress, pitch and duration. [12]

There are limits to how much speech recognition can influence speech synthesis or vice-versa. Fundamentally, these are two separate and distinct problems. For instance, speech recognition must be established to work for various speakers who all have voices of varying pitches and uses different stresses. On the other hand, speech synthesis has only one steadfast source of input: text. It simply needs to produce one accurate acoustic output. Additionally, recognition and synthesis techniques depart on signal processing means. Speech recognition mostly ignores prosody and relies heavily on mel-cepstral processing. In speech synthesis, mel-cepstral processing has been proven to have a low efficacy rate while prosody – the patterns of stress and intonation in a language – is essential. [12]

Despite the fact that there are a variety of groups working to improve speech synthesis and recognition, there does not appear to be many free applications designed to help and encourage communication. Several applications, such as Ventrilo [11], advertise themselves as a more convenient means of communication for gamers. Other applications are used for primarily commercial purposes, such as telephone calls from doctor's offices, prescription offices, and other automated services. The application created in this research is devoted simply to communication whether it is between those who are visually and hearing impaired, used as a transcriber, or as a teaching aid in learning phonics.

## **4. Implementation**

### **4.1 GUI**

As a primary concern for of the application is to increase the ease in which the visually and hearing impaired can communicate, it was essential for a simple and straightforward GUI to be created. Gtkmm [4] is a C++ wrapper for GTK+, a popular GUI library that is often packaged with UNIX system. It provides a basic GUI toolkit with widgets, such as windows, buttons, toolbar, etc. While the primary development for the application is taking place in MSVC++, there has been an effort to remain open for the possibility of cross platform capabilities. Gtkmm meets this by ensuring cross-platform on such systems as Linux (gcc), Solaris (gcc, Forte), Win32 (gcc, MSVC++ .Net 2003), MacOS X (gcc), and more.

At the current phase, a simple GUI exists, which displays widgets for the basic text input and networking capabilities. There are two basic GUI views that exist according to whether or not the user is connected to a server. The first view serves primarily as a text editor. It consists of a window with one text area in which text can be inputted either through the keyboard or (in the future) using speech recognition. This text can then be saved on to the computer. As long as the user is disconnected, text files may also be opened and will be displayed in the text area.

Using the “Options” choice under the Connections menu, the user (client) is able to specify the server’s IP address, port, and choose when to connect or disconnect. After the IP and port have been entered, the user then chooses connect. If the address is correct and the server is running, then the user will connect. The GUI reflects this occurrence. Rather than one window, three windows will no appear: one to type in, one showing the conversation, and another showing the names of the users who are currently online. Three buttons are also now available, which allow the user to send the text, clear his or her text area, and close the application.

### **4.2 Networking**

In addition to incorporating speech recognition and synthesis, the application acts as a chat room to allow for communication over long distances. Thus, a basic client/server network was necessary. SDL\_net [5] fit the needs precisely. SDL\_net is a small, sample cross-platform networking library that uses the much larger C-written SDL (Simple DirectMedia Layer). The aim of SDL\_net is to allow for easy cross-platform programming and simplify the handling of network connections and data transfer. It accomplishes this through its simple and portable interface for TCP and UDP protocols. With UDP sockets, SDL\_net allows half-way connections; binding a socket to an address and thus avoiding filling any outgoing packets with the destination address. A connectionless method is also provided.

The current implementation in the application uses SDL\_net to open a socket which waits for a user to connect. Once the user connects, a test message is sent. If it is

successfully received, then the client's GUI changes to reflect that it has successfully connect. At this stage, the server simply closes once the message is received. As the conclusion section of this paper will explain, this is not the final status of the networking as chat is impossible if the server merely sends a message then disconnects.

### **4.3 Speech Synthesis**

The Festival Speech Synthesis System was chosen to accomplish the task of speech recognition. Festival provides a general framework for multilingual speech synthesis systems, although it has the most features with English. It is useable through a multitude of APIs, including C++, Java, Emacs, and through SCHEME command interpreter. [2] Three specific classes of users are targeted: speech synthesis researchers, speech application developers, and the end user. As such, Festival is open source although many of the alterations can occur through its functions rather than altering of its code.

Festival was built on a Unix system and has been most thoroughly tested in such platforms. However, there is a Windows port that has had minor testing on MSCV6 and Cygwin[14] – which provides a Unix-like environment for machines which run on Windows. As the application was being built on MSVC7, there were several difficulties encountered. First, while it does have a port for MSVC6, one still requires a Linux environment to first compile and make the VCFMakefile. Cygwin provided the mediator from tar to MSVC projects. Secondly, Festival contains several deprecated code. For example, it includes the deprecated unistd.h rather than unistd. While Visual Studios should default to the old libraries and allow the deprecated use, this does not work in conjunction with the GUI and networking, which use the newer libraries.

Festival has been successfully run using either Cygwin or the Visual Studio's Command Prompt. Overall, the basic commands are quite simple. For example, the command (*SayText "Hello world."*) successfully produces the spoken words "Hello world." Festival also comes with the ability to directly read a text file with only a few commands. A variety of voices are available, which allows the user to find the most "natural" sounding voice to him or her. Furthermore, any produced speech may also be saved as a sound file for future use.

### **4.4 Speech Recognition**

Although the application was coded and primarily tested on Windows XP, an effort has been made to maintain the possibility for easy cross platform capabilities. Sphinx-3 continues this trend as it is workable on GNU/Linux, UNIX variants, and Windows NT or later. Written in C++, Sphinx-3 is a product of Carnegie Mellon University and is well known in the speech recognition community for its large vocabulary and, relatively, timely results. It includes both an acoustic trainer and various decoders (e.g. text recognition, phoneme recognition, N-best list generation). [15]

Sphinx-3 uses phonetic units to determine and build a word pronunciation. Two types of output may be produced: a recognition hypothesis and a word lattice. The

recognition hypothesis consists of the best recognition result for each utterance processed. A word lattice provides a list of all possible word candidates that were recognized during the decoding of an utterance. [15] It is very useful for determining whether an utterance was clearly spoken or not as, in theory, the utterance should appear somewhere in the word lattice although it may not appear in the recognition hypothesis.

In order to test that it has successfully been installed and allow a sample program of how to incorporate it, Sphinx-3 comes with a sphinx3-simple.bat, which allows the user to practice its speech recognition abilities using a limited vocabulary. A user who has never used a speech recognition program or has not used the SphinxTrain yet can expect 30-40% successful speech results. One early test of the recognition process involved stating, "Start recording. One two three four five six." This produced "START THREE CODE R E Y TWO THREE FOUR I SIXTH" as Sphinx-3's hypothesis as to what was spoken. For the most part, it is evident where these mistakes may have come from. "SIXTH" is very close to "six." In fact, the added "-th" is, most likely, a misinterpretation of the ending pause and possible background noise that was picked up. Sphinx-3 attempts to eliminate silences, coughs, and white noise, but it is not always perfectly accurate. Additionally, "THREE CODE" vaguely resembles "record." "R E Y," however, does not resemble any portion of recording or one, thus exemplifying the imperfection of the untrained recognition system.

Besides for occasional misrecognized words, Sphinx-3 has several other limitations. First of all, without a segmenter, Sphinx-3 cannot be used in utterances longer than 300 seconds. Secondly, Sphinx-3 does not recognize capitalization. All spoken words are transformed into entirely capitalized hypotheses. Thus, a separate grammar correction portion of the application will be necessary to ensure correct capitalization. Punctuation is also omitted. However, it may be possible to tamper with the filler penalty file, which differentiates stutters or silences from words, such that prolonged pauses signify periods.

## 5. Conclusion

Vigorous searching of the internet produced a variety of applications or packages which incorporated speech synthesis and recognition. However, few seemed to incorporate both. Furthermore, some had not been update for several years or had limited testing or documentation and thus were not particularly useful in studying the technologies and developing the application. The successfulness of such applications (as measured by the accuracy rating) is currently mediocre in normal conditions. Under optimal conditions (often requiring quiet isolation) significant increases of accuracy ratings were observed in speech recognition. Speech synthesis had high accuracy; however, many of the noncommercial voices still have a computerized, abnormal sound with slight imperfections in pitch, intensity, or duration.

The application created in this project uses two popular speech recognition and synthesis packages – Sphinx-3 and Festival, respectively – to ensure accurate and well-tested results. Basic networking allows for a simple text chatting. Conversion using recognition or synthesis occurs at the user end and is toggled by the user so that either feature may be turned off if they are not needed. While it has currently only been tested on Windows XP, all of the packages and implementations have been tested on Linux systems, and, thus, should be cross-platform with little, if any, changes.

Through this application, it is hoped that a simple solution is provided to communication between the hearing and visually impaired that is free and does not require any additional equipment besides a computer. Additionally, it is hoped that this application may also be used in educational settings, regardless of students' or teachers' disabilities, as a teaching aid. Thus, it may be employed for web-based education purposes.

## 6. Future Work

There are several aspects of the GUI that have yet to be incorporated. Currently, the GUI does not yet contain a means to choose whether or not speech recognition, speech synthesis, or both are currently being used. An options window or additional controls need to be added to allow the user more control over what speech aspects are being used. Without such a GUI feature, a hearing disabled person may be unnecessarily using the speech synthesis part and wasting virtual memory or RAM. Additionally, a window needs to be created to allow the user to choose a screen name and create a profile so that chat members are differentiated. There are several other GUI features that would be useful but not necessary: choosing of font attributes stored with each user's profile, a datafile which stores all of the preferences, the ability to block users, private conversations, user-specific voices, and volume control.

Through the GUI, the user can now attempt to connect to a server. If the server is active, the user receives a small message (currently, a "welcome" string) and then disconnects. An inactive, or incorrectly inputted IP address, results in an error message being displayed in the text window. Currently, chatting does not occur as the client disconnects and the server closes as soon as the message is received. This must be altered so that the server remains up and users are able to chat amongst themselves.

The speech recognition and synthesis packages have been successfully installed and tested through their respective sample programs that come with them. Next, these packages must be implemented with the GUI and networking. This will involve some minor changes with the Festival package as a few deprecated lines of code are used. Additionally, the dictionaries of both Sphinx-3 and Festival must be edited to include common chat lingo, emoticons, and text that, outside of a chat room setting, would be considered gibberish. Currently, the speech recognition aspect has an accuracy of approximately 30-40%. This is quite common as no official training has occurred. Through the Sphinx trainer, this accuracy rate can drastically be increased to over double its current rate.

It is also essential that rigorous testing occurs in order to optimize the usability of the application created in the research. Dr. Karen Cole, the director of the Academic Support Office at Stetson University, has offered her assistance throughout the process. She also commented that she would be willing to contact students with hearing and visual impairments so that usability may be tested by the key users of the application. Additional testing will occur among those without impairments to ensure that everyone can use it regardless of impairments.

## APPENDIX A: SPEECH GLOSSARY

### **Grapheme**

A grapheme is the smallest part of written language that represents a phoneme in the spelling of a word. A grapheme may be just one letter, such as b, d, f, p, s; or several letters, such as ch, sh, th, -ck, ea, -igh [16].

### **Phone**

A phone is speech-sound considered as a physical event without regard to its place in the sound-system of semantics of a language. A set of phones is called a phoneme [17].

### **Phoneme**

A phoneme is the smallest part of spoken language that makes a difference in the meaning of words. English has about 41 phonemes. A few words, such as "a" or "oh", have only one phoneme. Most words, however, have more than one phoneme: The word if has two phonemes (/i/ /f/); check has three phonemes (/ch/ /e/ /k/), and stop has four phonemes (/s/ /t/ /o/ /p/). Sometimes one phoneme is represented by more than one letter [16].

### **Phonics**

Phonics is the understanding that there is a predictable relationship between phonemes (the sounds of spoken language) and graphemes (the letters and spellings that represent those sounds in written language) [16].

### **Syllable**

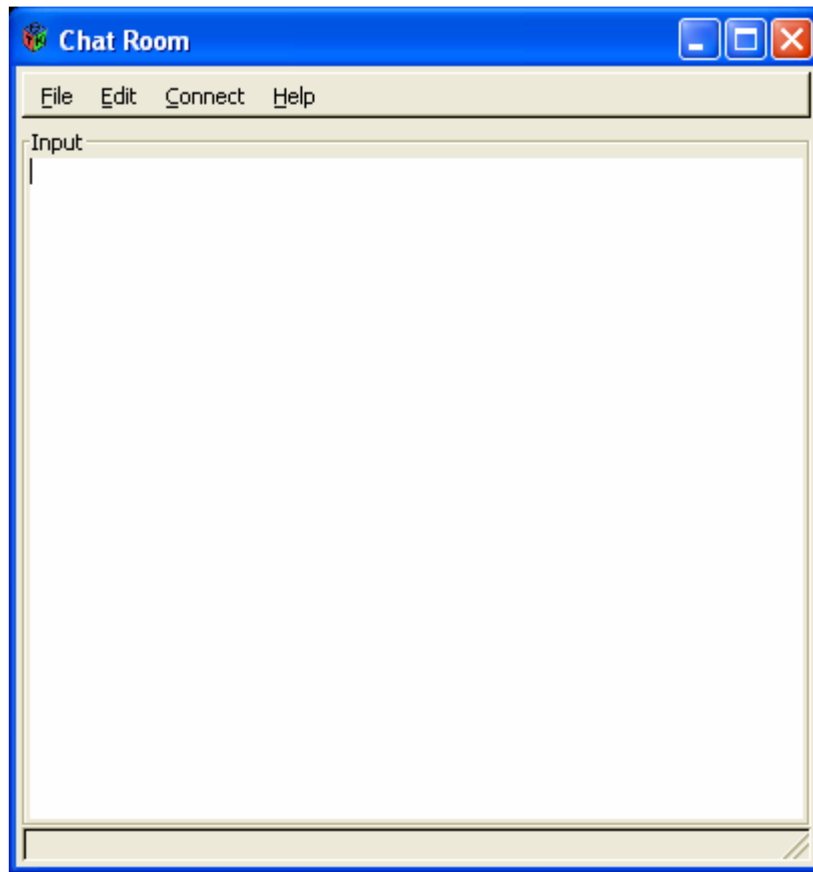
A syllable is a word part that contains a vowel or, in spoken language, a vowel sound (e-vent; news-pa-per; ver-y) [16].

### **Utterance**

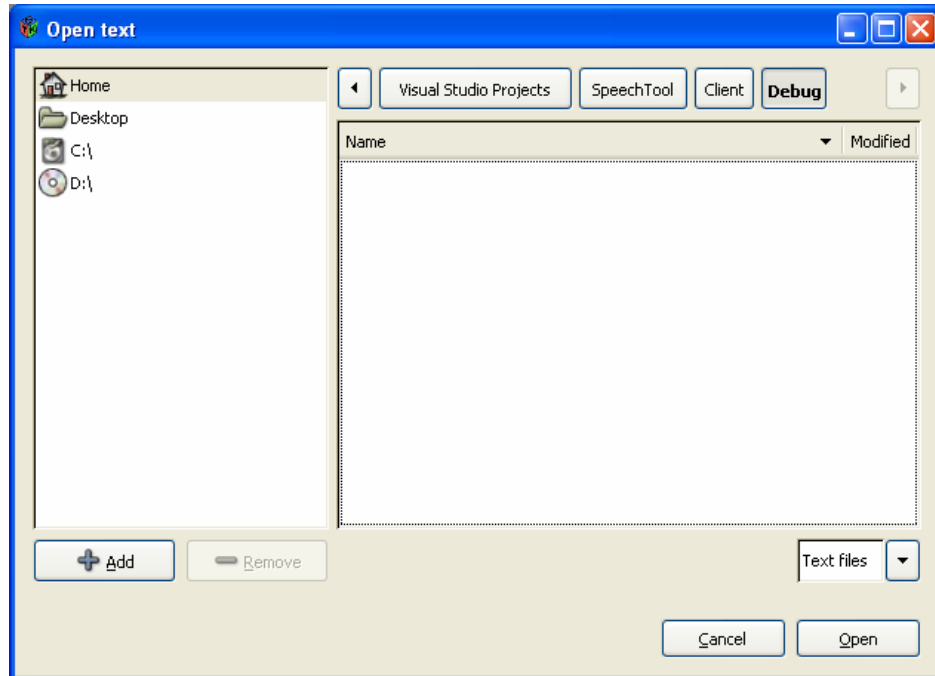
An utterance is a complete unit of speech in spoken language. It is generally, but not always, bounded by silence [18].

## APPENDIX B: APPLICATION SCREENSHOTS

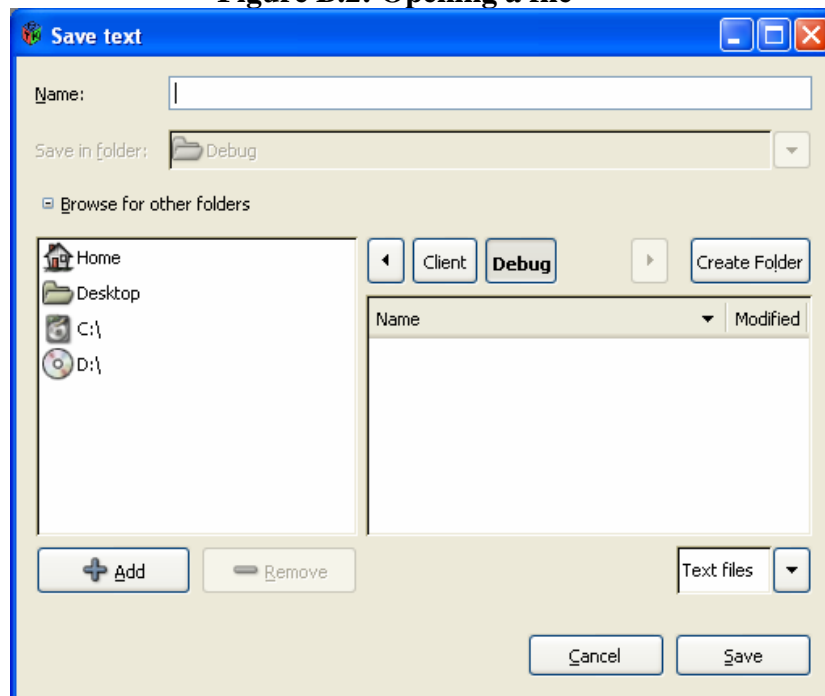
### B.1 Disconnected User Images



**Figure B.1: Disconnected User Window**

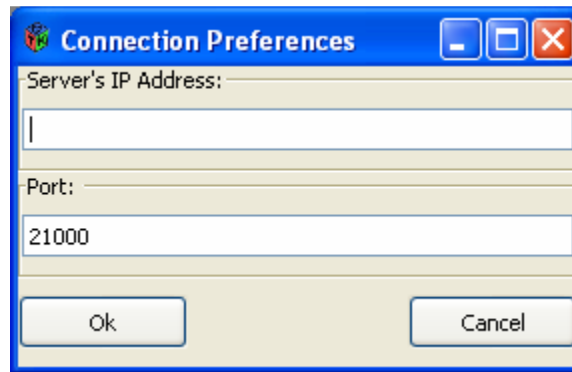


**Figure B.2: Opening a file**

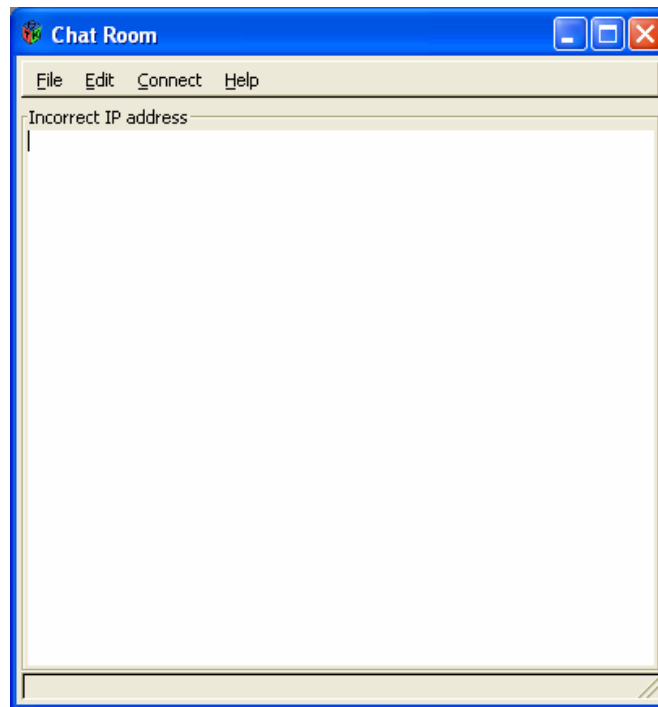


**Figure B.3: Saving text to a file**

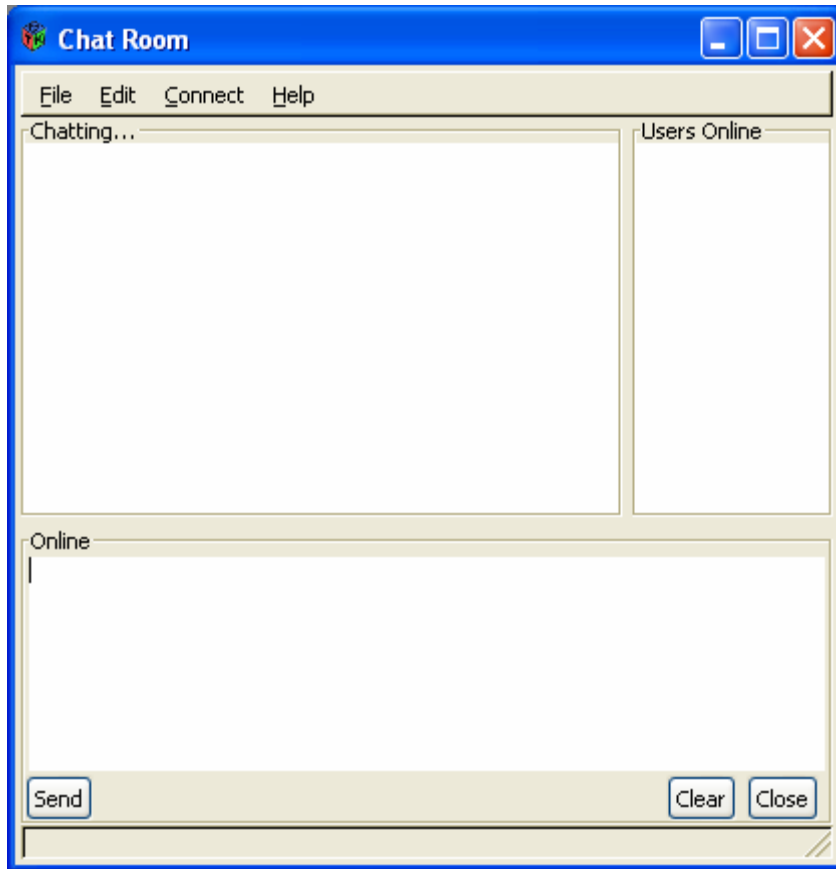
## B.2 Connected User Images



**Figure B.4: Connection Preferences**

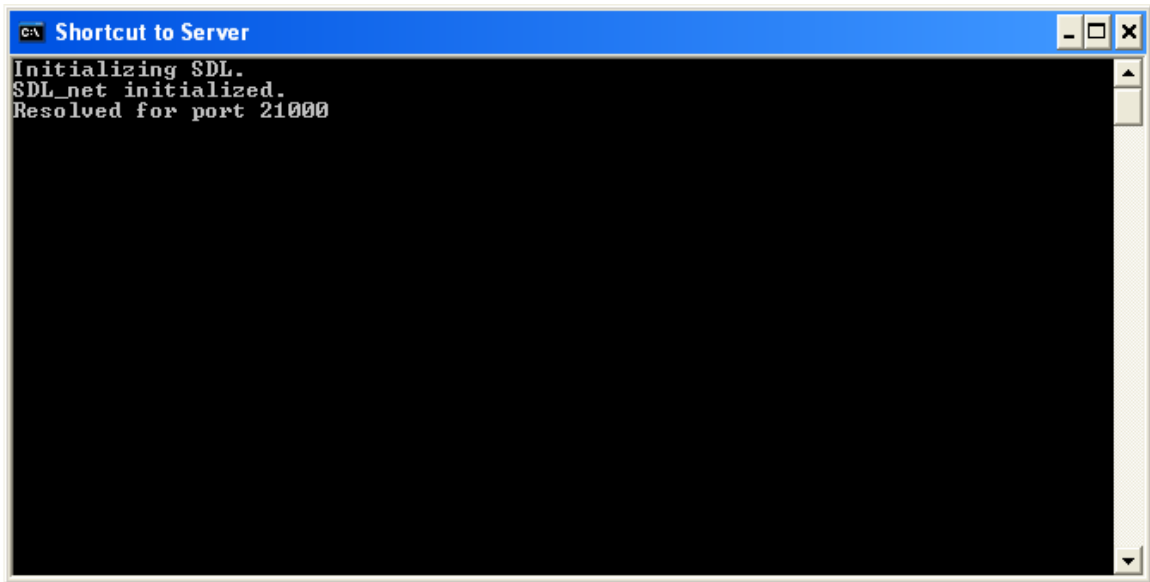


**Figure B.5: Failed to connect because of IP**



**Figure B.6: Connected Client**

## B.3 Server Images



```
CA Shortcut to Server
Initializing SDL.
SDL_net initialized.
Resolved for port 21000
```

Figure B.7: Initialized server

## REFERENCES

- [1] Freedom Scientific, "FSTTY Deaf-Blind Solution Pricing," [Online], Available: [http://www.freedomscientific.com/fs\\_products/fsttypricing2.asp](http://www.freedomscientific.com/fs_products/fsttypricing2.asp)
- [2] Black, Adam, et al., "Festival Speech Synthesis System," [Online], Available: <http://www.cstr.ed.ac.uk/projects/festival/>
- [3] "The CMU Sphinx Group Open Source Speech Recognition Engines," [Online], Available: <http://cmusphinx.sourceforge.net/html/cmusphinx.php>
- [4] Cumming, Murray, et al. "Gtkmm – the C++ Interface to GTK+," [Online], Available: <http://www.gtkmm.org/>
- [5] Lantinga, Sam, Masahiro, Minami, and Wood, Roy. "SDL\_net Documentation Homepage," [Online], Available: [http://jcatki.no-ip.org/SDL\\_net/](http://jcatki.no-ip.org/SDL_net/)
- [6] "Speech Synthesis," [Online], Available: [http://en.wikipedia.org/wiki/Speech\\_synthesis](http://en.wikipedia.org/wiki/Speech_synthesis)
- [7] "TTS FAQ," [Online]. Available: <http://www.research.att.com/projects/tts/faq.html#TechWhat>
- [8] "Speech Recognition," [Online], Available: [http://en.wikipedia.org/wiki/Speech\\_recognition](http://en.wikipedia.org/wiki/Speech_recognition)
- [9] Brusilovsky, Peter. "Adaptive and intelligent technologies for web-based education". *Special Issue on Intelligent Systems and Teleteaching*, (Kunstliche Intelligenz), 4:19-25, 1999.
- [10] "Speech Research Lab," [Online], Available: <http://www.asel.udel.edu/speech/ModelTalker.html>
- [11] "Ventrilo – Surround Sound Voice Communication Software," 2005. [Online], Available: <http://www.ventrilo.com/>
- [12] Bulyko, Ivan and Ostendorf, Mari. "The Impact of Speech Recognition on Speech Synthesis," 2002. *Proceedings of 2002 IEEE Workshop on, 2002*. Available: [http://crow.ee.washington.edu/people/bulyko/papers/TTS02\\_invited.pdf](http://crow.ee.washington.edu/people/bulyko/papers/TTS02_invited.pdf)
- [13] A. Le *et al.*, "The 2002 NIST RT Evaluation Speech-to-Text Results," *Proc. RT02 Workshop*, 2002. Available: <http://www.nist.gov/speech/tests/rt/rt2002/>
- [14] "Cygwin Information and Installation," [Online], Available:

<http://www.cygwin.com/>

[15] Ravishankar, Mosur K. "Sphinx-3 s3.X Decoder (X=5)," [Online], Available: <http://cmusphinx.sourceforge.net/sphinx3/>

[16] "IEL. Live Chat. Glossary," [Online], Available: <http://www.illinoisearlylearning.org/chat/marks-glossary.htm>

[17] "Phones," [Online]. Available: <http://en.wikipedia.org/wiki/Phones>

[18] "Utterance," [Online]. Available: <http://en.wikipedia.org/wiki/Utterance>