SEES – AN ADAPTIVE MULTIMODAL USER INTERFACE FOR THE VISUALLY IMPAIRED

by

APARAJIT SAIGAL

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2007

ACKNOWLEDGEMENTS

To my wife - I am deeply indebted to you. My Parents, who nudged me along as always. To my siblings, for pushing me to get another degree. To my sons, who thankfully did not enroll in college, before I graduated.

Special thanks to Professor Levine for his guidance and ideas and to my committee members, for signing the papers.

April 13, 2007

ABSTRACT

SEES – AN ADAPTIVE MULTIMODAL USER INTERFACE FOR THE VISUALLY IMPAIRED

Publication No. ______

Aparajit Saigal, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: David Levine

The enormous amount of electronic data present today can be a daunting task to access and process for a regular person, let alone someone with a disability. The World Wide Web and Electronic Mail have transformed the way we live. We are constantly become more dependent on this information, communication and commerce medium.

The Speech Enabled Email System (SEES) is an alternate user interface that allows for the retrieval of emails and RSS feeds using speech. SEES can be accessed via a desktop application or through a telephony interface such as a regular phone-line or

Voice over IP. This thesis explores the different types of interfaces, describes SEES in length and compares SEES with interfaces that attempt similar functionality.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. World Wide Web and Email

In today's information age, there are millions of terabytes of data residing over networks and many more being generated every day. E-mails are becoming a part and parcel of everyday communication. The extensive data stored on the World Wide Web is becoming the repository for any information one ever wanted. Traversing and siphoning relevant information from this conglomerate of data is still an extensive task for anyone, particularly not having a disability, let alone for someone with one.

Statistically, one in ten of the world's population has a disability of some kind during their lives [2]. Technology exists which helps people with disabilities accomplish computer-related tasks in a more efficient manner. Text-to-Speech, voice recognition, specialized hardware for other types of data input and retrieval exist today to accommodate physically and sensory disabled members of society. The advance in these technologies is now enabling software developers to design interfaces to all this information, to satisfy the needs of people with limited abilities.

1.2. Human Computer Interfaces for the Disabled

Traditional Human Computer Interfaces (HCI) can accommodate only a certain population of computer users. With the advance in mobile computing and as Weiser [4] calls the third wave of computing – Ubiquitous Computing, the need for more than justa keyboard and mouse is becoming essential. HCIs are evolving from the windows-icons-menus-pointers (WIMP) interfaces to sophisticated multimodal interfaces.

"An executive often doesn't want to use the traditional keyboard because it is slow and awkward, while a physically disabled person can't use it because of some physical limitation. Both need quick efficient access to computer systems with minimal effort. However, access problems are accentuated by a physical disability when an alternative input mode is essential rather than a feature. It may turn out that developments for helping disabled persons will have an impact upon the able-bodied population and help the business executive" [3].

Therefore as Dr. Busby [2] mentions even if we develop technology for the disabled not for pure altruism or for the fulfillment or one's professional accountability, then the customer base of close to three hundred million people, along with the wealthy executives mentioned above should be a good financial motivator to proceed with developing such technologies.

Humor aside, the power to better another's life by using knowledge one possesses, generates a sense of fulfillment. The opportunity to extend the basic rights to the disabled community with the help of technology, which in turn improves their lifestyle, is a task and a responsibility that should be taken seriously by the people of the scientific community.

The brilliant mind of Stephen Hawkings would never have unleashed the theories of black-holes if we had turned our backs on this community, other disabled people would have not had the freedom to enjoy a vocation, if technology had not intervened. This would lead to a social calamity as well, with the increase of tax-payers money now paying for the care of this community. We should help them as far as possible to become independent. Freedom to do what they desire, would in turn grant them the full citizenship that democracy promises.

Numerous interfaces already exist today for the disabled. There are modified keyboards, voice controlled interfaces, gaze tracking interfaces [8] etc. that will allow users with specific disabilities to operate a computer. These interfaces have already helped millions of disabled and given them the opportunity to be employed, to be social and to belong to society [6]. Advances in this field are essential to the betterment of society and lifting the stigma that a disability may hold.

1.3. Speech Enabled Email System - SEES

Focusing on a single disability, namely the blind or visually impaired, there are not many interfaces filling the gap that can supplement speech as an alternate mode of input and output for this community. Speech synthesis has come a long way since it's inception, however, speech recognition, a more complex task, still requires more research to be truly used as a sole method for interfacing with computers and applications.

This being acknowledged there may be hesitance in the application development community to develop speech based interfaces. SEES is a system based on speech recognition that draws on novel techniques in the realm of speech interfaces. It provides multimodality and ubiquity by provisioning alternate modes of input, output and access to the interface using telephony respectively.

The architecture of SEES allows for abstracting the speech recognition layer, concentrating on the use of techniques and methodologies for the speech interface. The underlying technology or Speech Recognition Engine therefore can be simply swapped out, with the emergence of new and better technology.

The practical aspect of the application is best relayed by quoting a phrase that a user may utter in their interaction with SEES - “New emails from Mom referring to Birthday form Yesterday”. The system can process this oral query and provide the corresponding result. This can be achieved by either sitting at a desktop with a microphone or dialing in via a telephone. All functionality of the application remains seamless from one modality to the other.

SEES is aimed at providing a beneficial service to the blind and visually impaired, but it’s approach to filtering techniques for speech interfaces and emails can prove to be useful and beneficial to others as well.

1.4. Thesis Structure

The thesis is organized into chapters dealing with the basic definitions and examples of certain types of Human Computer Interfaces. This is covered in Chapter 2. Chapter 3 looks at systems that accomplish some of the functionalities of SEES. Chapter 4 describes the technologies such as Automatic Speech Recognition that are used for SEES. Chapter 5 goes into details about the application of SR and TTS technologies along with telephony hurdles that were faced during the development of SEES. Chapter 6 provides details of the functions of SEES and exposes the architecture and provides Use Cases for the system. Finally Chapter 7 draws some conclusions and defines future work.

CHAPTER 2

HUMAN COMPUTER INTERFACES

There are many types of Human Computer Interfaces that have been developed that are revolutionizing the way we interact with computers. There are techniques used that make our experience with the current hardware interfaces that deliver a much more enjoyable, efficient and in the case of the disabled, a useful and usable interface. This chapter describes with examples the concept of Multimodal and Adaptive interfaces.

2.1. Multimodal Interfaces

Multimodal refers to using more than one modality. Multimodal computer user interfaces therefore are HCIs that allow more than one form of interaction with the computer. These interfaces rely on the advance in the field of recognition technology. Technologies such as voice recognition, pen input, vision tracking are being integrated into sophisticated interfaces to applications that try and recognize naturally occurring human behavior and language [13].

The additional modality can be used to supplement another mode, or serve as an alternate mode for convenience. An interface that extends speech recognition and keyboard input is an example of a multimodal interface. The aim of these interfaces is to mimic naturally occurring forms of human language and natural behaviors.

Literature defines input modes to such interfaces as Active or Passive [13]. Active input modes are ones that are deployed by the user intentionally as an explicit command to a computer system (e.g. speech). Passive input modes refer to naturally occurring user behavior or actions that are recognized by a computer (e.g. facial expressions, manual gestures). They involve user input that is unobtrusively and passively monitored, without requiring any explicit command to a computer. Blended multimodal interfaces incorporate system recognition of at least one passive and one active input mode (e.g. speech and lip movement systems).

Interfaces that incorporate passive input modes are continuously fed information from that passive mode. The need to distinguish when to process inputs from any other mode, not as a complementary input but as a unimodal input, becomes a primary concern for such interface developers.

Individual differences and abilities account for a preference of one modality over another. Introduction of more than one modality to an interface makes this field important for the disabled, since it gives a user with the lack of a certain ability of interaction, another option, at the same time sharing a common user interface.

2.1.1. The Media Room

The pioneering system, “The Media Room”, developed by the Architecture Machine Group at MIT in the late 70’s [19], demonstrated the power of these multimodal interfaces. The ability to point to an object on a screen, and simultaneous use speech to control the movement of that object, opened up the vast arena of today’s virtual reality simulations and games.

To make this possible the system relied on speech and spatial recognition technology. The DP-100 Connected Speech Recognition System (CSRS) by NEC

(Nippon Electric Company) America, Inc. was used by the system and could process up to 5 words or utterances per sentence. It could use both the Dictation and Command and Control mode, though the vocabulary that it could recognize, since it had to store it in "active memory", varied from 120 words for the former to 1000 words for the latter.

Spatial Data was "recognized" by a space and orientation sensing technology developed by Polhemus Navigation Science, Inc., of Essex, Vermont. The system, called ROPAMS (Remote Object Position Attitude Measurement System) which based it's measurements made on a "nutating magnetic field".

The setup of the room, as seen in Figure 2.1, shows the Media Room with a chair that has a sensor pad on the left arm and a joystick on the other. The screen in the front of the room displays the projected image of a computer screen.

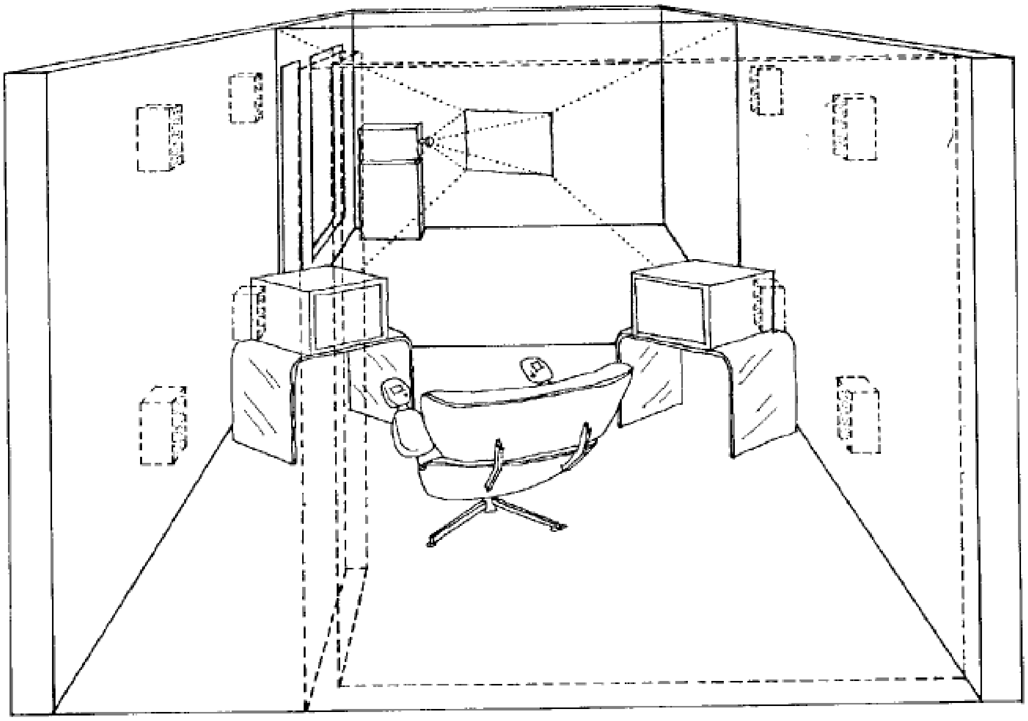


Figure 2.1 Sketch of the Media Room [24]

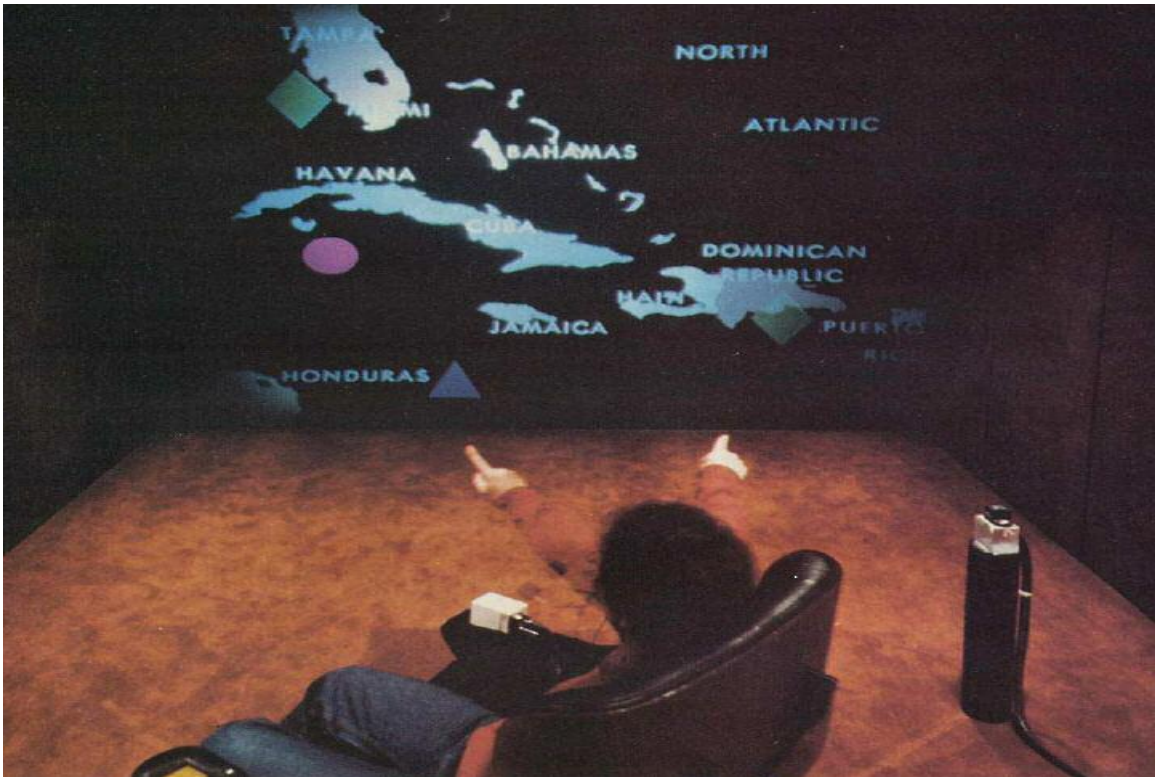


Figure 2.2 Put-that-there [24]

As seen in figure 2.2 the user points to the object he wants to move and could utter the words "Put that there", and the object pointed at is moved to the appropriate part of the screen.

2.1.2. Multimodal Myths

A few myths summarized from Oviatt [9] highlight the misconceptions that the general public may have about multimodal systems.

Myth #1: *If you build a multimodal system, users will interact multimodally*

Myth #2: *Speech & pointing is the dominant multimodal integration pattern*

Myth #3: *Multimodal input involves simultaneous signals*

Myth #4: *Speech is the primary input mode in any multimodal system that includes it*

Myth #5: *Multimodal language does not differ linguistically from unimodal language*

Myth #6: *Multimodal integration involves redundancy of content between modes*

Myth #7: *Individual error-prone recognition technologies combine multimodally to produce even greater unreliability*

Myth #8: *All users' multimodal commands are integrated in a uniform way*

Myth #9: *Different input modes are capable of transmitting comparable content*

Myth #10: *Enhanced efficiency is the main advantage of multimodal systems*

These misconceptions should be duly noted when designing a multimodal interface. As expanded by Oviatt, in myth #10, the advantages of a Multimodal Interface, apart from enhanced efficiency, include, amongst others, the accommodation of a wide range of users, tasks, and environments - including users who are temporarily or permanently handicapped.

Development of such interfaces is becoming essential in this day and age of handheld personal computers and web-enabled telephones. All the tasks one can accomplish using a WIMP interface, cannot be done using the same methods on these ever shrinking devices. Mobile use of devices in itself can be associated with a state of temporary disability. For example a person driving may not be able to use the stylus on a PDA, but speech provides a useful option to let the user transition from a stationary to mobile use.

2.2. Adaptive User Interfaces

An adaptive user interface is a software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user [10]. This definition implies that this is not a standalone system and its adaptability is rooted in the interaction with the user, allowing it to learn, predict, and in turn improve future interactions with the user.

The goal of predicting interests of a user is the same goal as that of machine learning [10]. These two methodologies are rooted in similar techniques of achieving this result. They both rely on a knowledge base, a learning environment and a performance task on which this learning should lead to improvement. The main differences between machine learning and adaptive interfaces is that the

- The knowledge base in machine learning is the data set used by the learning algorithms where as the user model is the knowledge base in an adaptive interface.
- Learning environment in machine learning are algorithms where as it is the user in an adaptive interface.
- Interactions become the performance task on which learning should lead to improvement in an adaptive interface.

Therefore, rather than relying on machine learning algorithms that rely on large datasets to form their knowledgebase for their training, adaptive interfaces rely on the user itself to train the system. Adaptive Interfaces use information filtering (IF) techniques to direct users to items/data, from a large set of items/data, that they may find interesting. The two types of commonly used IF approaches are:

- **Content-based filtering**: Content-based filtering techniques recommend items to the user by comparing the content of the item with the user's preferences.

- **Social/Collaborative filtering:** uses the rating patterns of similar users. It is based on the assumption that people with similar rating patterns will like the same kind of information. A collaborative filtering technique then uses the input of a "like-minded" user base and creates a profile to help in the filtering process.

To use these IF techniques profiles must be created for the users. The user model acts as the filtering rule for this information. Essentially, the user is modeled and profiled using one of the following techniques [11]

- **User-Created Profile:** This is the most simple and natural approach. The user specifies his area of interest by a list of terms. These terms now constitute his profile, which is then used in the filtering process.
- **System Created Profile by Automatic Indexing**: A set of data items which have already been judged by the user as relevant, are analyzed in order to identify the most frequent and meaningful terms in the text. Those terms, weighted according to the frequency of their appearance, constitute the user profile.
- **System and User Created Profile:** This methodology combines the above approaches. First, an initial profile is created using automatic indexing. Then, the user reviews the proposed profile and updates it, by adding or deleting terms, and changing their weights.
- **System Created Profile based on Learning by Artificial Neural Network:** Based on a sample set of data items that have already been

judged relevant by the user, an Artificial Neural Network (ANN) may be trained. The inputs of the ANN are the meaningful terms, and the outputs are the relevance judgments of the users. After training, the ANN serves as the user profile for future filtering.

- **User-Profile Inherited from a User-Stereotype**: This method assumes that the IF system has pre-defined user-stereotypes. A user-stereotype is represented as a content-based profile. A user-stereotype is also represented by a set of demographic and social attributes that are common to those users. A new user is attached to a predefined stereotype to which he/she is most close with respect to the demographic and social attributes.

2.2.1. Adaptive User Interfaces

The above mentioned filtering techniques are extended to recommendation systems. Some commercial applications of this technology are successfully being used today by e-commerce websites, news-on-demand services etc. that direct users to information that it predicts might be useful to them. Listed below are two of the better known sites, amongst countless others, that use the concept of an Adaptive Interface to promote their business and better the user experience:

- ***LAUNCHcast.com* -** An internet radio site, a subsidiary of Yahoo, provides the user an option to rate songs, artists or albums. These ratings are stored in the user's profile. Based on these ratings the site then plays other songs that the user may like. This prediction is based on the user's profile and profiles of users that have a similar rating of the songs picked by the user.

- ***Amazon.com* -** A search of a book, or any other item sold on the site, guides you to recommendations of other books that you may find interesting. This again is based on the ranking system that the site offers it's users.

2.3. Multimodal and Adaptive User Interfaces for the Disabled

The interfaces described in this chapter, which offer to give users a personalized experience, but more essentially prevent information overload, can help provide the disabled with a better computing experience as well. A blind person does not want to sit in front of the computer – navigating the screen using cumbersome voice commands searching for information.

Decreasing the time spent on a computer by filtering/personalizing the content for a user improves the user's productivity. These Adaptive methodologies promote the idea of giving the fewest instructions to the computer to retrieve the most useful content, which prove to be a significant step towards the improvement in HCIs for the disabled community.

Making the WIMP interface Multimodal by making it voice activated will help the blind person use the computer, but if the voice commands were merely simulating the mouse or the keyboard the interface does not prove to be useful.

To apply these methodologies for helping the disabled, the disability itself must be kept in the forefront of the design specifications. There is no single interface that will satisfy or compensate for all disabilities; interfaces for the disabled have to be tailored to the specific disability. However, the combination of Adaptive and Multimodal interface concepts can provide the general backbone in the development of these specialized interfaces.

CHAPTER 3

HUMAN COMPUTER INTERFACES FOR THE DISABLED

As described in the previous chapter the research into the different types of interfaces and sensory technologies can definitely help the disabled with their interactions with a computer. Whether it be an Adaptive Interface that profiles the user's activities, helping for example a mentally challenged person with tasks on the computer or it be a multimodal interface that may helps a person with a vision impairment use their voice, these interfaces if implemented with the disabled in mind, could make a significant difference in the disabled community.

In the following subsections of this chapter the previous concepts will be exemplified by exploring some of the technologies that either are helping or could potentially help the disabled.

Section 3.x will focus specifically on applications for the blind and visually impaired with emphasis on accessibility of information, specifically emails and web-pages. This section will help identify the need for the development of the key features of SEES. Though there are a number of types of assistive technology products and applications [39], [40], only those mentioned in section 3.x were the ones experimented with.

3.1. Assistive Technologies

According to the Wikipedia [44] Assistive Technology (AT) is a "generic term that includes assistive, adaptive, and rehabilitative devices and the process used in selecting, locating, and using them." By providing enhancements or alternate methods of interaction, AT promotes greater independence for people with disabilities by enabling them to perform tasks that they were formerly unable to accomplish.

Narrowing down the Assistive Technologies to the ones that help a disabled user interact with a computer, the categories of devices listed below are available and used by the disabled to enable their interaction with a computer [40]. Examples of TTS, Speech Recognition and Screen Magnifiers – some of the tools used by the blind are not mentioned as they are explained in further detail with examples in the next section.

1. **Alternate Input Devices –** Devices that are not the standard keyboard and mouse for inputs to the computer. A few examples are:
   - **Alternative keyboards** – keyboards that have larger or smaller keys, alternative key configurations or are designed for use with one hand.
   - **Electronic pointing devices** – for manipulating the cursor on the screen without using hands with technologies such as ultrasound, infrared, eye movements [8], nerve signals or brain waves.
   - **Sip-and-puff systems** - inhaling or exhaling lets the user activate these systems.

- **Wands and sticks** - an alternate for the use of hands to press keys on the keyboard, these devices can be worn on the head, strapped to the chin or held in the mouth.
- **Trackballs** - movable balls on top of a base that can be used to move the cursor on screen.
- **Touch screens** – an alternate for the keyboard and mouse for the ease of selection of items on the screen. They can be either be built in to the monitor or overlaid on top of an existing monitor

2. **Braille embossers** – used for printing Braille from programs on the computer. A program has to interpret and direct this information to the embosser.
3. **Light signal alerts** - an alternate to alert the user with light signals other than sounds. Warning sounds or alerts are displayed as a light signal rather than a sound, which can benefit a deaf user.
4. **On-screen keyboards** - provide an image of a standard or modified keyboard on the computer screen that allows the user to select keys with any regular or alternate input devices mentioned above. This technology is helpful for individuals with dexterity or mobility difficulties.
5. **Refreshable Braille displays -** provide Braille output of information represented on the computer screen line by line. A program such as JAWS provides the positioning of the line to be sent to the display. A Braille "cell" is

composed of a series of dots. These dots, usually small rounded plastic or metal pins, are mechanically lifted or depressed as needed to form Braille characters.

6. **TTY/TDD conversion modems** are connected between computers and telephones to allow an individual to type a message on a computer and send it to a Teletypewriter (TTY)/ Telecommunication Device for the Deaf (TDD) telephone or other Baudot equipped device. The Baudot code is a character set predating EBCDIC (Extended Binary Coded Decimal Interchange Code) and ASCII . It was used originally, and is now used primarily, on teleprinters [44].

3.2. Internet for the Blind and Visually Impaired

The following subsections focus on the technologies that are available to help the blind interface with computers. The technologies mentioned were examined with the intent of finding out what the blind are using and what they may find helpful for either accessing email or the World Wide Web.

3.2.1. Microsft XP Language toolbar and Microsoft Outlook

Microsoft's email client, Microsoft Outlook is compatible with the Windows Office XP's Language toolbar shown in figure 3.1. The Language toolbar, interfaces with the Office products enabling them with Speech Recognition. The toolbar is generic and operates in a *dictation* or *voice command* mode, concepts that are reviewed in Chapter 5.

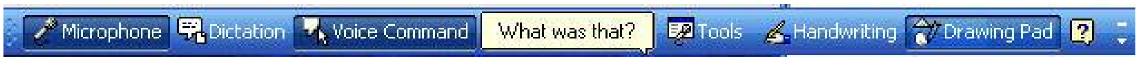


Figure 3.1 Microsoft XP Language toolbar

The language toolbar, if not already active, can be initialized from Tools menu using the Speech option as seen in figure 3.2.

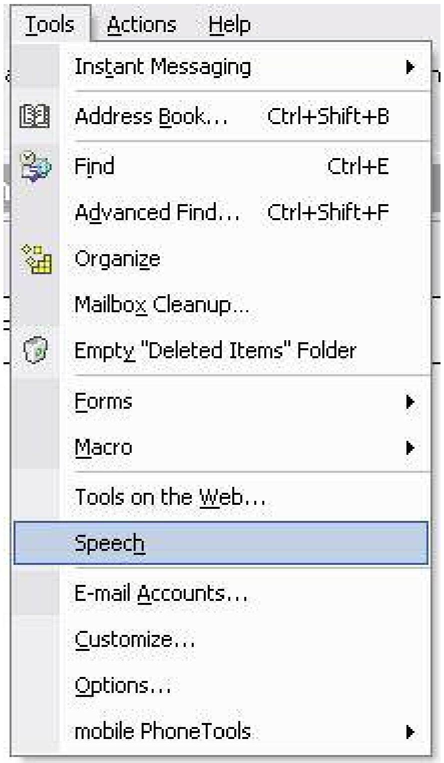
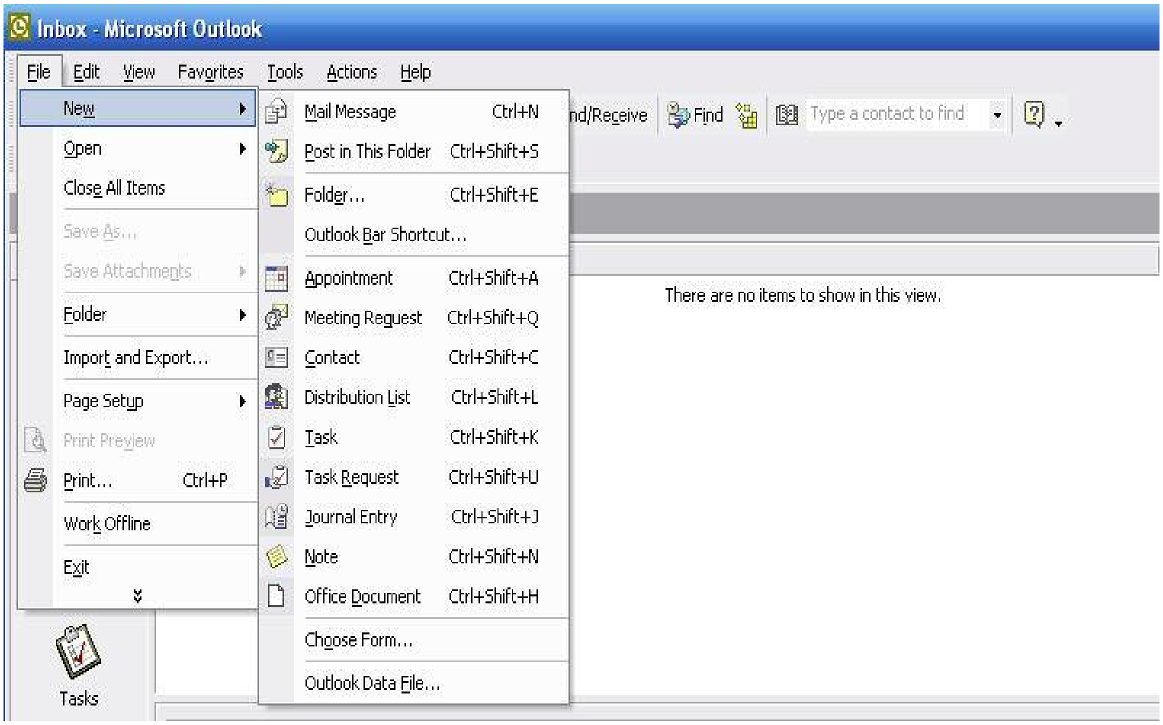


Figure 3.2 Activating the language bar

Depending on the application, i.e. Word, Excel, Outlook etc. the Voice Command portion of the application adapts itself to the commands specific to the application. These commands are applicable for the Menu bar of the application.

As an example saying “File” would bring up the File menu of the Outlook application. Any item within the File menu shown in figure 3.3 can then be activated by subsequently speaking it’s label, such as Mail Message, in our example.



There are no items to show in this view.

Figure 3.3 Outlook “File...New”

Switching to the dictation mode by saying “dictation”, one can then proceed to dictate an email. The speech interface works well with Microsoft Outlook, and most commands or functions can be navigated to, however the speech toolbar, does not give a reference point, or read aloud the sub-menus of the menu window it opens.

This would deter a blind person from using only the speech interface for Outlook; however a visually impaired person may be able to use the speech interface, in conjunction with a screen magnifier such as MAGIC or the accessibility options found in Windows to magnify the screen.

3.2.2. JAWS, Microsoft Outlook, Internet Explorer

JAWS is the leading screen-reader product made by Freedom Scientific that is very popular amongst the blind community. It provides speech-synthesis capabilities to any window/program on the screen. It acts as a generic reader with the ability to read text on any window that is opened up. This includes menu windows or sub-windows as well.

The benefits of this screen-reader utility when combined with Windows software that provide shortcuts to their menu items becomes an extremely powerful tool for the blind. Navigation using the keyboard becomes easy using the keyboard as the user has a reference point via the auditory responses and alerts provisioned by JAWS.

Microsoft Outlook with virtually all it's menus accessible by shortcuts, proves to be a strong ally with JAWS. The numerous shortcuts provided, a small example seen in table 3.1, allow you to access most, if not all, functionality that is provided by the GUI. With JAWS reading every screen and alert that opens up, the user can use the keyboard for any data entry, if required, or navigate efficiently through the various screens to perform tasks such as reading and sending emails.

Table 3.1 Sample JAWS and Microsoft Outlook Shortcuts

Description	**Shortcut Key**
Send Message	CTRL + ENTER OR ALT + S
Reformat an email message from RTF to plain text	CTRL + SHIFT + O
Delete Message from message window	CTRL + D
New Contact Dialog	CTRL + SHIFT + C
New Office Document	CTRL + SHIFT + H
Read Warning Header	CTRL + INSERT + W
To Save Non-Email Item in Current Folder	ALT + S
Cancel the current operation	ESCAPE KEY
Move up current level of treeview	UP ARROW
Move down current level of treeview	DOWN ARROW
Collapse current branch of treeview	LEFT ARROW
Expand current branch of treeview	RIGHT ARROW

There World Wide Web Consortium (W3C) has an entire methodology called the "Web Content Accessibility Guidelines" [43] that web designers can follow to make their web pages more accessible. The methodology promotes practices such as minimal frames use of <alt> tags consistent use of Header tags etc. This helps screen readers, such as JAWS to efficiently use the HTML itself for navigation purposes e.g. JAWS let's you traverse from heading to heading, taking advantage of the <H1>, <H2> HTML tags.

Table 3.2 Sample JAWS and Internet Explorer Shortcuts

Description	**Command**
Back a Page	ALT+LEFT ARROW or BACKSPACE
Forward a Page	ALT+RIGHT ARROW
Move to Address Bar	ALT+D
Read Address Bar	INSERT+A
Move JAWS Cursor to Address Bar	INSERT+A twice quickly
Virtual HTML Features	INSERT+F3
Activate Mouse Over	INSERT+CTRL+ENTER
View Basic Element Information	INSERT+SHIFT+F1
View Advanced Element Information	CTRL+INSERT+SHIFT+F1
Move to Next Clickable Element	SLASH
Move to Previous Clickable Element	SHIFT+SLASH
Select Clickable Element	INSERT+CTRL+SLASH
Move to Next Mouse Over Element	SEMICOLON
Move to Previous Mouse Over Element	SHIFT+SEMICOLON
Select a Mouse Over Element	INSERT+CTRL+SEMICOLON

JAWS and Microsoft Internet Explorer perform well together. Taking advantage of the shortcut keys provided by Internet Explorer, the web pages are read aloud, including the alternate or <alt> HTML tags that are provided with images. There are shortcuts for navigating from tab to tab in the new versions of Internet Explorer and JAWS is fully compatible with tabbed browsing as well.

3.2.3. VoMail

A paid service VoMail provides a telephone interface to the internet by providing Text-to-Speech access to

- **News:** News summaries by voice.
- **Blogs:** Blogs read over by voice.
- **RSS Feeds:** Listen to web (RSS) news feeds.
- **Sports:** Scores, schedules, rankings, and statistics for your selected sports teams.
- **Stocks:** Five portfolios with today's prices and total portfolio values.
- **Weather:** Current and 5-day forecasts for up to six different cities.
- **Email:** Email messages read to you – no navigation currently provided.
- appu2821@ifbyphonemail.com,(866) 203-8700 now

On setting up an account with VoMail, the user gets to setup the preferences for the above services. A toll-free telephone number is provided for the use of VoMail. An email address is provided and is the one that is used with this service, an option to setup external email addresses for VoMail is not available.

Upon dialing the number the user is prompted to select from the above listed options. The information pertaining to the choice selected by the user is then read out. However, there is no barge-in capability and the user must listen to the entire text, whether it is email or RSS feed.

The service has no filtering capabilities for emails, and no navigational options to jump from one message to the other. Besides these shortcomings a blind or visually impaired person may be able to use this service, as the voice prompts and voice recognition work decently well.

3.2.4. Email2phone

Email2phone is a commercial service that offers to call your phone for every email that you receive. There is a cost associated with the service. The e-mail is read over the phone using Text-to-Speech. A reply can be sent to the sender by dictating it via the phone, when prompted.

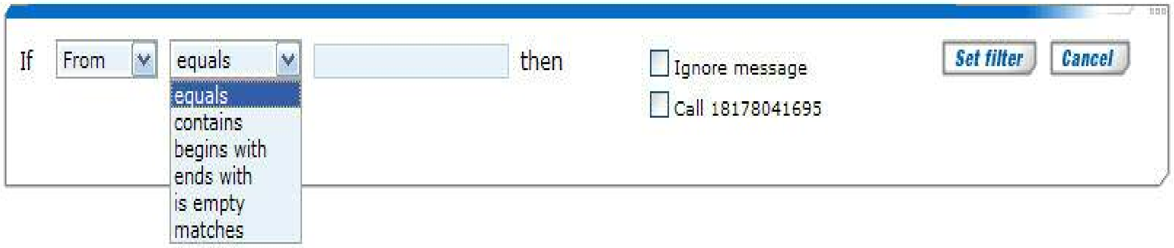


Figure 3.4 Email2phone filters [32]

The interface for setting up the service is web-based. The initial setup requires a username, password and a telephone number. Subsequently a filter on the body, from or subject fields of the email can be constructed. This allows for directing only emails of interest to the phone line.

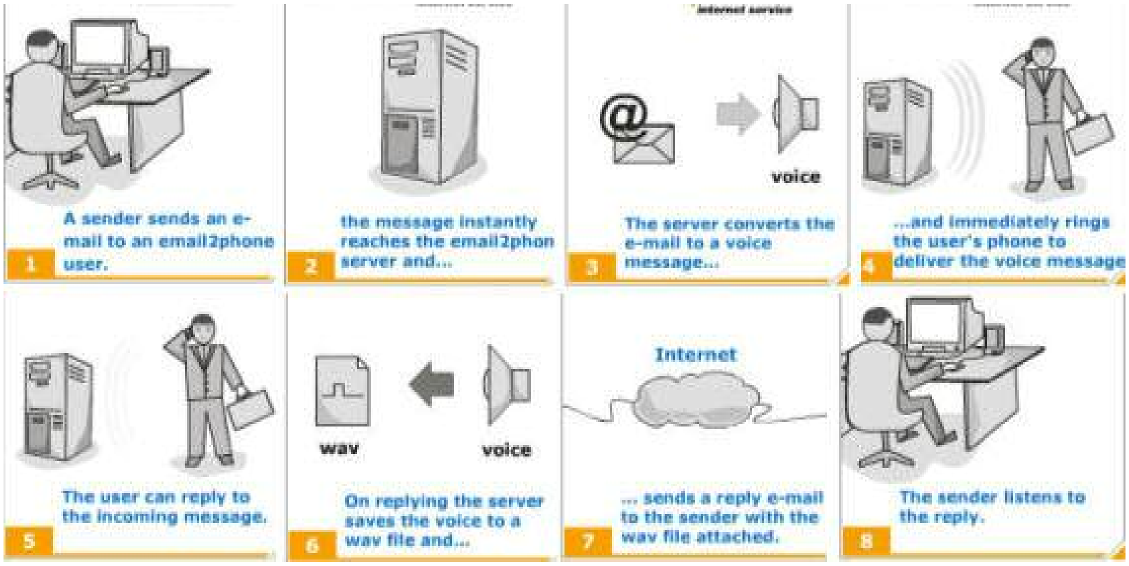

internet service
1
A sender sends an e-mail to an email2phone user.
2
the message instantly reaches the email2phon server and...
voice
3
The server converts the e-mail to a voice message...
4
...and immediately rings the user's phone to deliver the voice message
5
The user can reply to the incoming message.
wav
voice
6
On replying the server saves the voice to a wav file and...
Internet
7
... sends a reply e-mail to the sender with the wav file attached.
8
The sender listens to the reply.

Figure 3.5 Email2phone Application Flow

This service can be used for calling any telephone in the world; however the cost is based on that telephone number. Email2phone provides a unique service as seen in the illustration in figure 3.5.

CHAPTER 4

RELATED TECHNOLOGIES AND CONCEPTS

This chapter delves into the concepts such as Ubiquitous Computing, Automatic Speech Recognition, Text to Speech and Information Extraction. These technologies are the foundations on which SEES is built. The Chapter provides the theoretical prospects of these technologies; the practical application of these technologies is dealt with in Chapters 5 and 6.

4.1. Ubiquitous Computing

Marc Weiser of Xerox, the father of Ubiquitous computing who coined this term defines this as “calm technology”. He describes it as the third wave of computing – the first being the mainframe where lots of people connected to a single computer, the second where the computer and user interact one on one and the third - ubiquitous - integrating information displays into the everyday physical world, where computing is in the background and invisible to the user[4].

Ubiquitous computing is sometimes confused with virtual reality but is quite the opposite. Virtual reality let’s people enter into a computer-generated world where as Ubiquitous Computing brings the computer into every aspect of the world. Information can be retrieved or transmitted to and from anywhere.

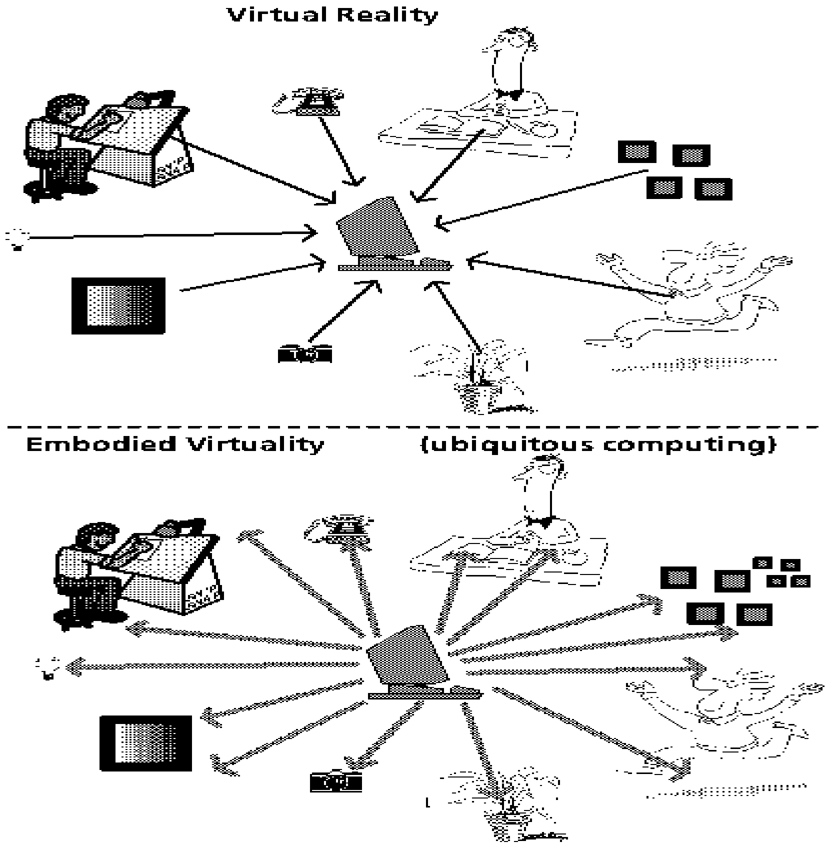


Figure 4.1 Virtual Reality and Ubiquitous Computing [16]

Not having to worry about which device to use in order to get or send information will free people of cumbersome computing devices that one has to interact with daily. The benefits to the disabled community will be a freedom to choose the method of interaction based on their abilities.

4.2. Information Extraction

The process of retrieving relevant information from unseen text is known as Information Extraction (IE). IE is not the same as Information Retrieval, which is the process of finding known text in documents. Whereas IR simply finds texts and presents them to the user, the typical IE application analyses texts and presents only the specific information from them that the user is interested in.

IE and IR can be best explained with the help of an example. Let's assume a person wants to go to the University of Texas at Arlington for graduate studies. She is interested in finding out the tuition, deadline for submitting her application for the Fall Semester and the living costs in Arlington.

Being the computer savvy person she is, she will type in the relevant keywords in Google which will proceed with it's IR process and give her a few hundred documents. She will now examine these documents herself and gather the information she was looking for from the sources.

An IE application written for such a query would have given our ambitious student all the information, by performing the analysis on the retrieved data, programmatically, giving her only the information she was interested in.

4.2.1. Types of Information Extraction

There are five types of IE (or IE *tasks*) as defined by the forum for this research, the Message Understanding Conferences (MUCS) [35].

1. **Named Entity recognition (NE)** - Finds and classifies names, places, etc.

2. **Coreference resolution (CO)** - Identifies identity relations between entities in texts.
3. **Template Element construction (TE)** - Adds descriptive information to NE results (using CO).
4. **Template Relation construction (TR)** - Finds relations between TE entities.
5. **Scenario Template production (ST)** - Fits TE and TR results into specified event scenarios.

IE plays a key foundation concept in the filtering process that is implemented in SEES as seen in Chapter 6. The Information Extraction process of SEES is achieved using the open-source Java framework known as GATE (General Architecture for Text Engineering). GATE is defined briefly in Chapter 5.

4.3. Automatic Speech Recognition

Speech is the transfer of information via acoustic signals. People encode information in speech sounds when talking, and decode those sounds when listening, in order to recognize the words spoken to them.

The set of speech sounds produced when talking are differentiable by their acoustic characteristics and is the primary property that is exploited by the Automatic Speech Recognition (ASR) systems, which match this signal with stored representation of speech units, hence recognizing the best match.

However, many factors, such as sex, age, race, physical fitness etc. produce variations in the acoustic characteristics of speech. Apart from these physical attributes the stature of a person in society, levels of education and emotions all have a major effect in the way speech is delivered.

Given these variations, the ability of humans to decode speech is impeccable. People have routine conversations in different dialects in a noisy environment.

We accomplish this by applying linguistic and world knowledge to the acoustic information provided by our sense of hearing. In order to better understand speech and language let's analyze their structure.

A language is be specified by a lexicon and a grammar. The lexicon is a dictionary of the words in the language; a grammar describes the language syntax, the ways in which words may be combined into sentences. Lexical units may be further decomposed into morphemes, indivisible grammatical or semantic units that may stand alone as words or be grouped to form other words, e.g. print-s, print-ed, print-ing, re-print [36]. Similarly, the unit of word construction in speech is the phoneme, "defined such that if one phoneme is substituted for another in a word, the meaning of the word may be changed" [42]. Only certain phoneme sequences are permissible in any given language.

Grammar also needs to be applied while decoding speech, though there is a difference between the sentential or textual grammar and the grammar of everyday verbal communication. Grammatical phrases, though, tend to be recognized by listeners where as ungrammatical language is not ordinarily spoken.

As an example one may understand the grammatically incorrect "him and me", but a sentence such as "him me and", with no grammar, is generally not spoken.

Semantic information, or the meaning of speech, contained in the conjunction of words and in contextual references, is also present in the decoding process. The application of these semantics clarifies ambiguous words that may not be differentiable at the acoustic or lexical levels. As an example without any semantics applied “hermit” and “her mitt” are indistinguishable at the acoustic level, but in when taken into context are very much recognizable in a sentence [17].

Human speech recognition takes into account all the above knowledge. A string of phones are heard, allocated to their respective phonemes. Words are constructed according lexical rules, and their parent sentences deduced from grammatical, semantic knowledge.

Speech recognition technologies allow computers equipped with a source of sound input, such as a microphone, to interpret human speech. Speech Recognition Engines (SREs) have been developed by several academic and commercial institutes to bring into realization, the fantasies of talking to a computer, and it doing your bidding. Though, still not as accurate as Kitt of Knight Rider – these engines are getting better every year.

The Speech Recognition process starts with the digital sound captured by the sound card through a microphone. The sound is converted into a more manageable format by a converter, which translates the stream of amplitudes that form the digital sound wave, into its frequency components. Phonemes, the elementary sounds that are the building blocks of words, are then identified. This is accomplished by mapping each frequency component of the sound to a specific phoneme. This process finishes the conversion from sounds to sentences.

Finally, a grammar, the list of words known to the program, lets the engine associate sets of phonemes with particular words, concluding the final step of Speech Recognition, which is to analyze the string. This analyzing applies semantics to the string to try and achieve a “sensible” recognition of the utterance. [14]

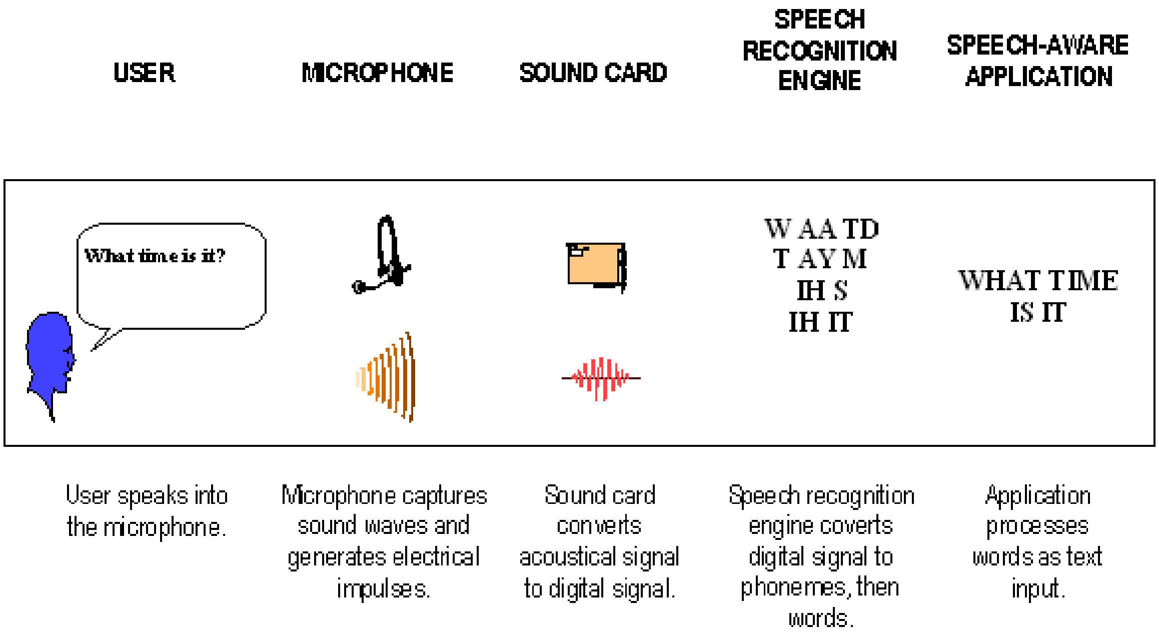

USER
MICROPHONE
SOUND CARD
SPEECH RECOGNITION ENGINE
SPEECH-AWARE APPLICATION
What time is it?
W AA TD
T AY M
IH S
IH IT
WHAT TIME
IS IT
User speaks into the microphone.
Microphone captures sound waves and generates electrical impulses.
Sound card converts acoustical signal to digital signal.
Speech recognition engine coverts digital signal to phonemes, then words.
Application processes words as text input.

Figure 4.2 Speech Recognition [15]

SR Engines can be categorized as Discrete or Continuous. In the former the user is required to pause between each word; in the latter the SR recognizes a more natural, continuous speech. Discrete SR Engines are now being replaced by the more sophisticated Continuous SR Engines. These two classes each can be divided into subclasses of user-dependent and user-independent, the former being easier to implement. Continuous SR Engines support two modes of interaction

1. **Dictation:** User continuously speaks to the computer and the SR attempts to translate the speech into text – like a stenographer. This mode requires an in-context grammar.
2. **Command and Control:** User can initiate certain actions using limited spoken commands based on a loaded context-free grammar.

A context-free grammar (CFG) defines a specific set of words, while an in-context grammar involves a virtually endless list of words. The voice patterns for a different user speaking the same set of words may differ, calling for the need of training the Speech engine to each user's voice. Therefore Dictation SR needs more processing power than Command and Control SR in which the only comparison done by the SR Engine is on the CFG.

4.4. Text-to-Speech

The process of converting text into spoken language is known as speech synthesis or Text-to-Speech (TTS). The synthesizer or TTS Engine, on a very high level, generates digital audio by breaking the word into phonemes, taking care of, amongst other things, special handling of numbers, punctuations etc.

TTS voices tend to sound less human than a voice reproduced by a digital recording, though research, such as that by AT&T on their Natural Voices product, is trying to bridge that gap and having great success at doing so.

These sophisticated engines are now providing features that produce both male and female voices and can synthesize multiple languages, accents, and can also be trained to pronounce words in certain ways if the original programming is not to your liking. The rate at which the words are spoken can be adjusted. Dictionaries can be loaded pertaining to accents from a certain region.

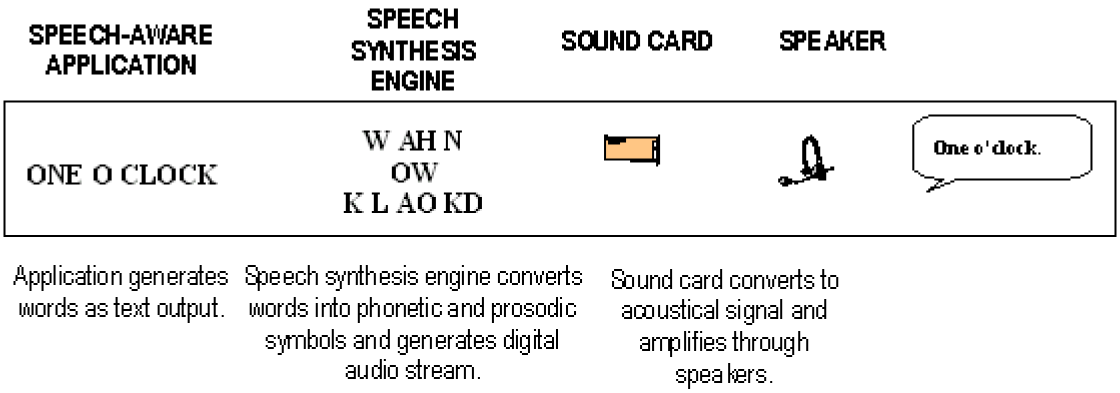

SPEECH-AWARE APPLICATION
SPEECH SYNTHESIS ENGINE
SOUND CARD
SPEAKER
ONE O CLOCK
W AH N
OW
K L AO KD
One o'clock.
Application generates words as text output.
Speech synthesis engine converts words into phonetic and prosodic symbols and generates digital audio stream.
Sound card converts to acoustical signal and amplifies through speakers.

Figure 4.3 Text-to-Speech [15]

CHAPTER 5

SAPI, TAPI, MODEMS AND TELEPHONY CARDS

This chapter delves into the basics of Speech Application Protocol Interface (SAPI) and Telephony Application Programmer's Interface (TAPI) pertaining to SEES.

The TAPI white paper [26] and Amundsen [27], describe in detail the internals of TAPI. For developing SEES the basic knowledge of the architecture as shown in figure 5.4 was absolutely necessary. The architecture illustrates the reasons behind some of the failures of certain modems and telephony boards that did not perform certain tasks that were required for the SEES architecture.

5.1. SAPI

The Speech Software Development Kit (Speech SDK) for SAPI developed by Microsoft was used in the prototype for Text-to-Speech (TTS) and Speech Recognition. SAPI is an interface specification that enables developers to create applications for a variety of Speech Recognition (SR) and Text-to-Speech (TTS) engines from different vendors. These engines need to be SAPI compliant in order for developers to write standard code for a Speech application regardless of the underlying engines.

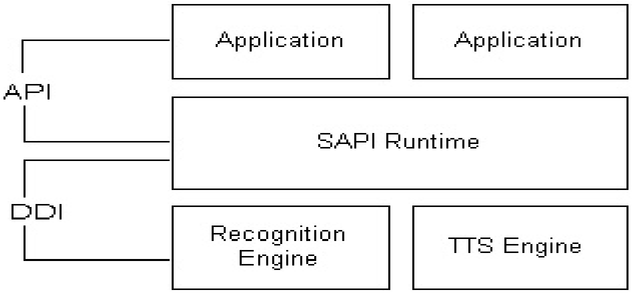
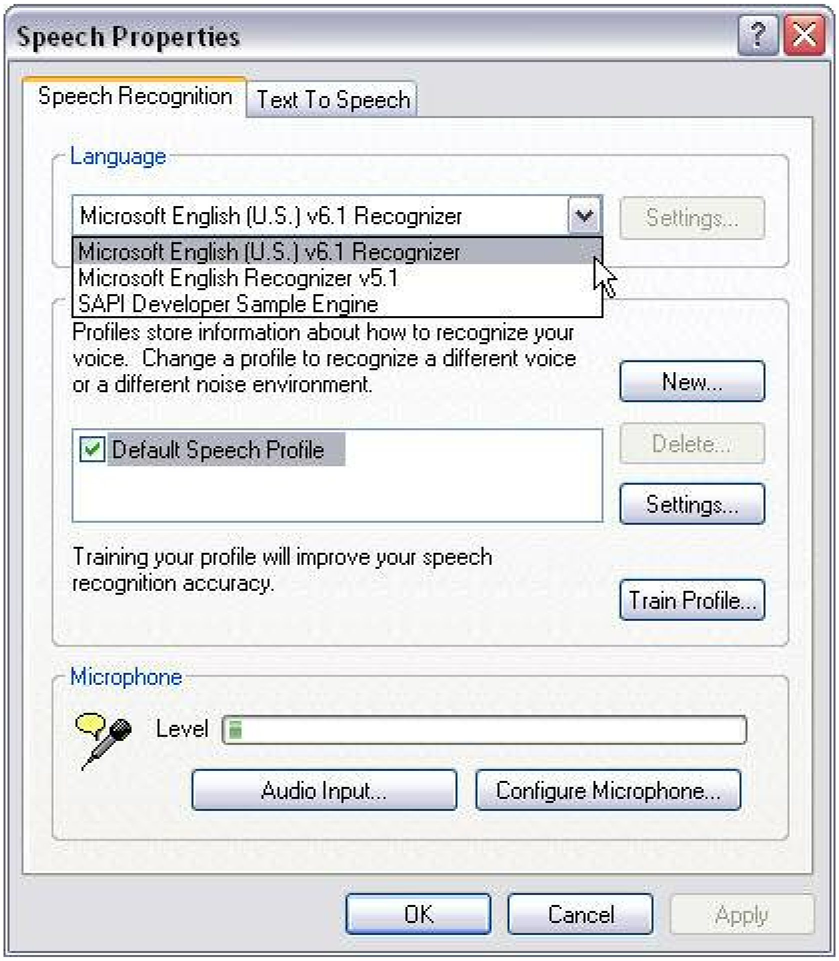


Figure 5.1 SAPI Architecture [15]

SR and TTS engines such as Dragon Naturally Speaking [28], and the open-source SPHINX [29] and Festival [30] projects at Carnegie Mellon University were evaluated during the process of this thesis. For development and prototyping purposes SEES uses the engines that ships standard with Windows XP.



Speech Properties

Language: Microsoft English (U.S.) v6.1 Recognizer

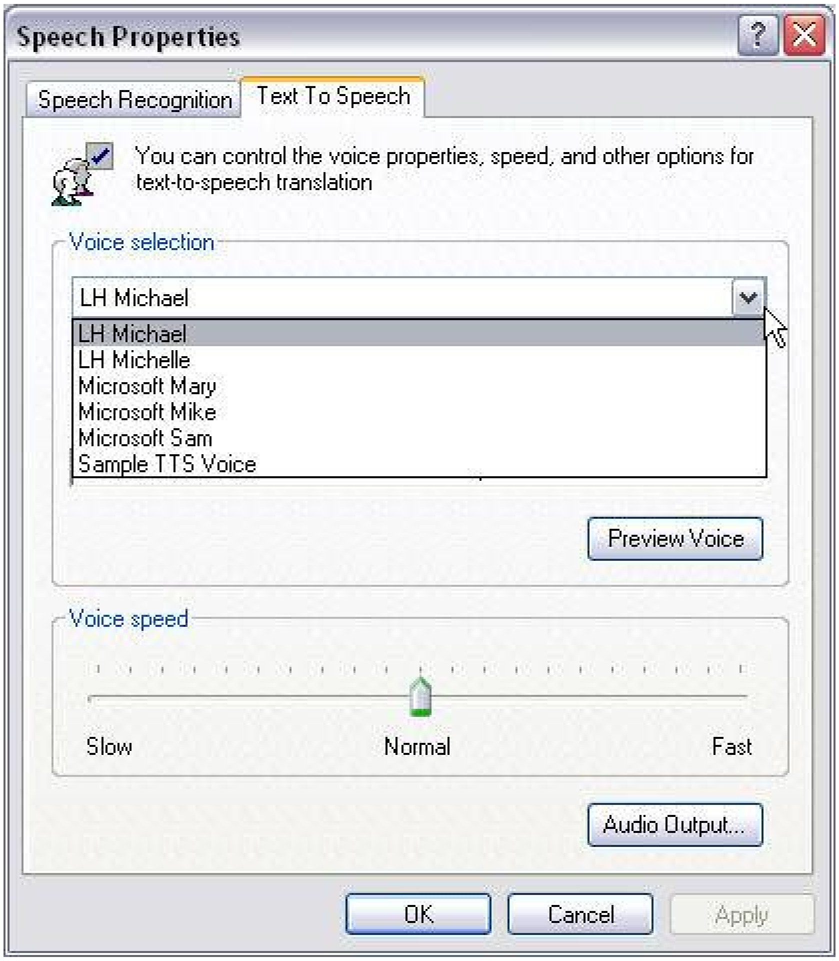
Profiles store information about how to recognize your voice. Change a profile to recognize a different voice or a different noise environment.

☑ Default Speech Profile

Training your profile will improve your speech recognition accuracy.

Figure 5.2 SAPI Speech Recognition Engine Selection

Besides the different algorithms used by the different speech synthesis engines, the voices that are used by these engines are configurable.



Speech Properties

You can control the voice properties, speed, and other options for text-to-speech translation

Voice selection

LH Michael

Figure 5.3 Microsoft Speech Recognition Engine Voice Selection

5.2. TAPI

TAPI is the model developed by Microsoft that provides an abstract layer for developers to access telephone services on the Windows Operating System. Telephony is a technology that integrates computers with telephone networks.

TAPI provides a significant advantage to developers as the code no longer depends on the hardware. As long as the hardware vendor provides a TAPI-compliant Service Provider Interface (SPI) or TSP, the code runs seamlessly from one piece of hardware to the other.

All functions and events that are possible with a regular phone line are available via TAPI. These include generation of all the dial tones, flash-hook, call answer, call forward etc. [26]

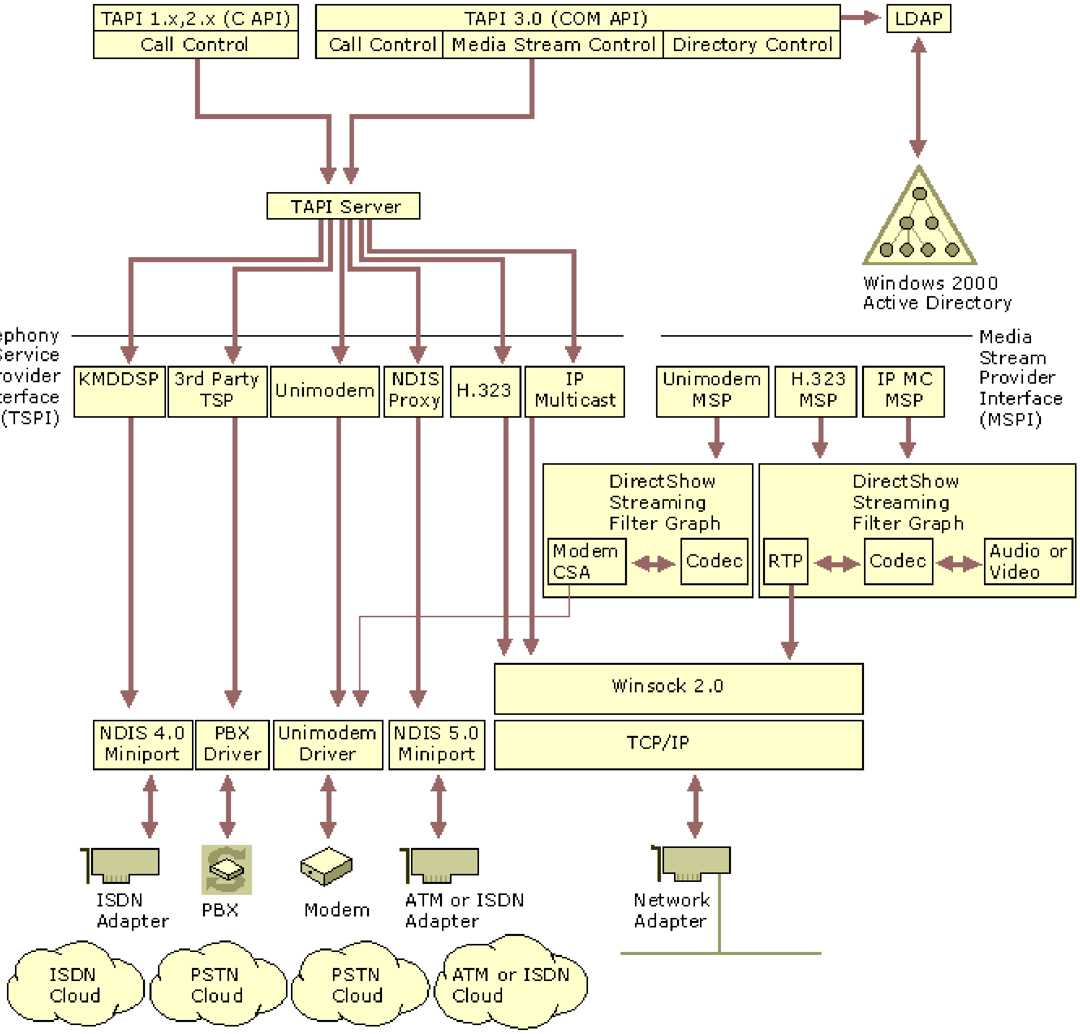


Figure 5.4 TAPI architecture [25]

The Microsoft H.323 Telephony Service Provider and its associated Media Service Provider (MSP) allow TAPI-enabled applications to engage in multimedia audio/video sessions with any H.323-compliant terminal on a local area network (LAN) or the Internet [25].

5.2.1. TAPI Devices

The TAPI design model is divided into two areas, each with it's own API, with each API focusing on a *device*. The two TAPI devices are

1. **Line device –** models the physical telephony line. The line device is a TAPI object representing the physical telephony line that provides a *handle* to the actual physical line.
2. **Phone device –** models the desktop handset. The model allows for the building of 'virtual phones' that take advantage of the components attached to the PC such as the sound card and microphone.

In short, the Phone Device connects to the Line Device which in turn connects to the actual physical telephony line.

5.2.2. Telephone Line Services

TAPI is designed to provide the same functionality on telephone lines regardless of physical characteristics. It provides a seamless interface on analog, digital or cellular lines. Line types can be divided into

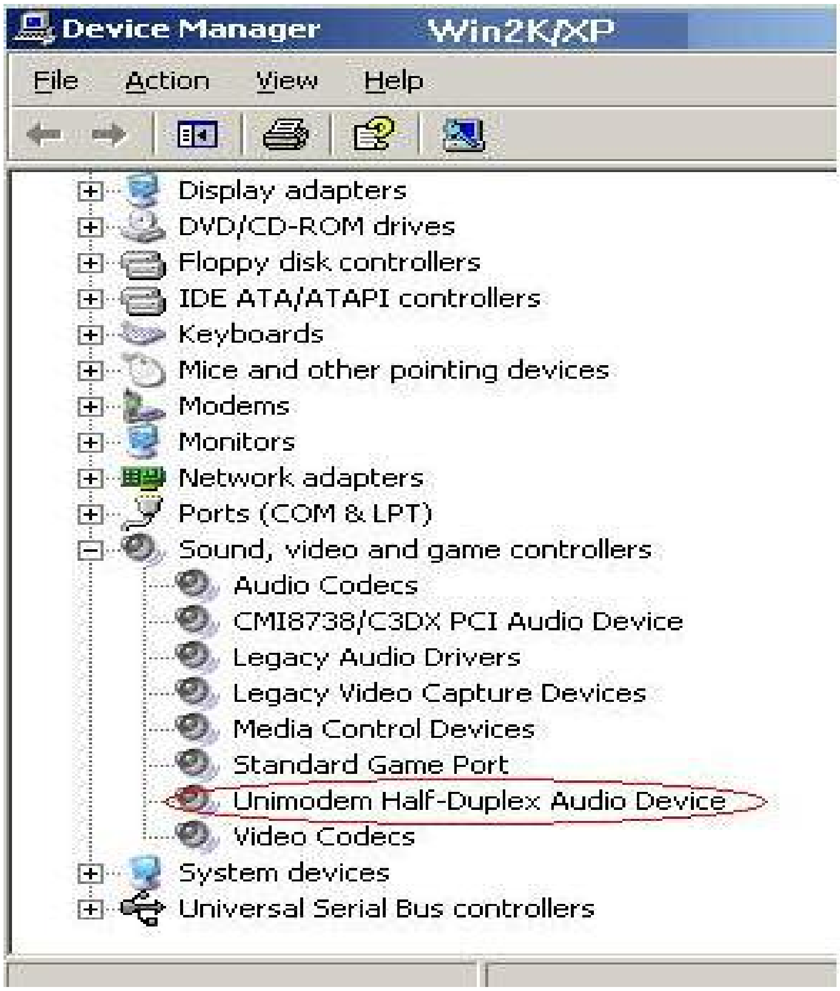
1. **Analog –** found in most homes.
2. **Digital –** used by large organizations e.g. T1 and ISDN.
3. **Private Protocol –** special kind of digital line, used within Private Branch Exchanges(PBXs) for voice, data and special control information

for switching hardware to provide features such as call-waiting, conferencing etc.

5.2.3. Modems and Telephony Cards

The difficulty in selecting a modem/telephony card proved to be two-fold. For one the TAPI protocol is not supported by all modem vendors. The technical information on the packaging, web-sites and sometimes even customer support was not clear on the full functionality and capability of the product. Even for Dialogic the industry's leader of telephony cards, TAPI support is sketchy.

Secondly, the Wave Device used by modems and telephony cards to send and receive audio to the sound card, via the system bus, is integrated into the Unimodem driver. As it turns out, Unimodem supports only half-duplex communication, as seen in figure 5.5. What this means is that there can be only one Wave Device open, either in or out, at a given time. Simultaneous communication required by SEES, where a *barge-in* command such as 'STOP' or 'NEXT', as seen in Chapter 6, is essential, cannot be accomplished. Either the application does Voice Recognition or Text-to-Speech, not both at the same time.



- Display adapters
- DVD/CD-ROM drives
- Floppy disk controllers
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Ports (COM & LPT)
- Sound, video and game controllers
  - Audio Codecs
  - CMI8738/C3DX PCI Audio Device
  - Legacy Audio Drivers
  - Legacy Video Capture Devices
  - Media Control Devices
  - Standard Game Port
  - Unimodem Half-Duplex Audio Device
  - Video Codecs
- System devices
- Universal Serial Bus controllers

Figure 5.5 Half–Duplex Audio Device in Win XP Device Manager

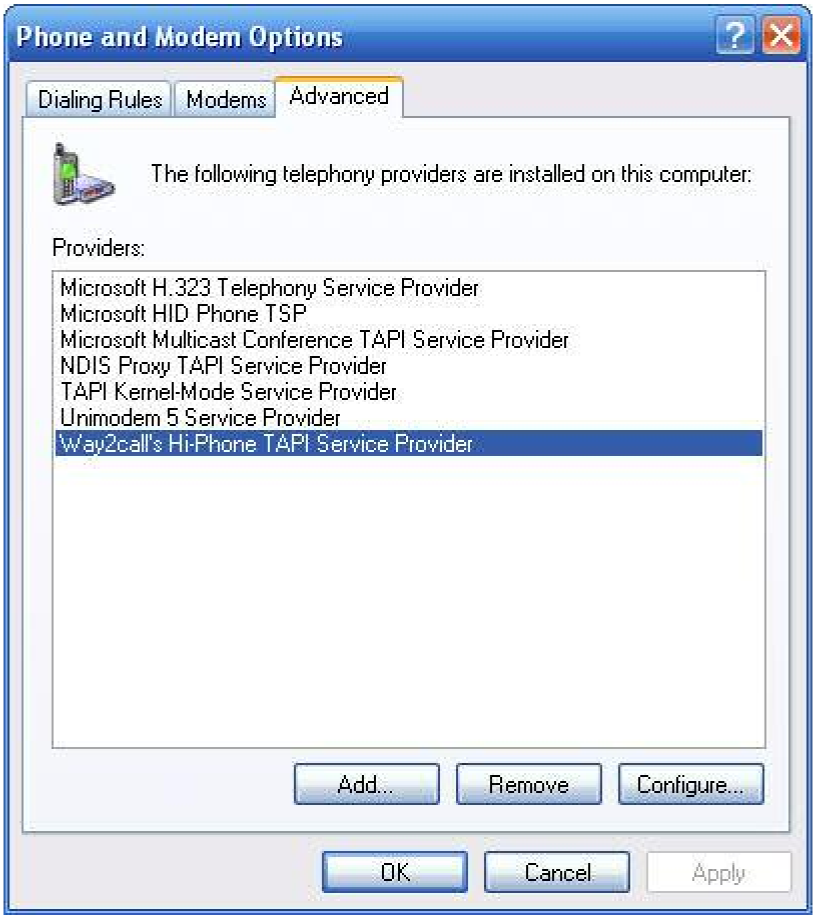
5.2.4. Tested TAPI Service Providers

The TSPs that were tested during the development of SEES were

1. **H323**.
2. **Unimodem** - Microsoft's TSP to all voice modems. This included testing
   a. Dialogic Telephony Board
   b. Several off the shelf voice modems

3. **Way2Call -** TSP and Way2Call Hi-Phone Desktop.

The TSP list is available in the settings of the Telephone and Modem options of the Control Panel as seen in figure 5.6.



The following telephony providers are installed on this computer:

Providers:

Microsoft H.323 Telephony Service Provider
Microsoft HID Phone TSP
Microsoft Multicast Conference TAPI Service Provider
NDIS Proxy TAPI Service Provider
TAPI Kernel-Mode Service Provider
Unimodem 5 Service Provider
Way2call's Hi-Phone TAPI Service Provider

Figure 5.6 Telephony Service Provider List

The Way2Call modem was the only hardware that provided it’s own TSP that is full-duplex. The device is fully TAPI compliant and worked as specified straight out of the box.

The H323 protocol worked as specified, and was used as the development of SEES, with the base definition of TAPI proving to be invaluable when switching from H323 to another line, in our case the Way2Call desktop. To test the H323 protocol the Microsoft Net Meeting Product [27] was used to initiate calls to the SEES telephony server described in Chapter 6.

Though the Dialogic telephony card could have been used for SEES, taking advantage of TAPI would not have been possible due to the lack of full-duplex TSP. The native driver and the proprietary interface provided by Dialogic would have to be used. The interface is supported only for the C++ language. The client code for SEES is written in Visual Basic, rewriting it in C++ would have been a major undertaking.

5.3. General Architecture for Text Engineering

A Natural Language Engineering (NLE) framework developed at the University of Sheffield, GATE is touted as the Eclipse [31] of NLE. It is a theory-neutral (i.e. many different views of the appropriate data structures to describe human language can be accommodated) framework for the management and integration of Natural Language Processing (NLP) components and documents on which they operate [5].

5.3.1. Language Engineering and Natural Language Processing

NLP is a branch of computer science that studies computer systems for processing natural languages. As Gazdar [33] mentions it includes the development of algorithms for parsing, generation, and acquisition of linguistic knowledge. This field of research and study includes:

1. The study of the feasibility and efficiency of theses algorithms.

2. Use and development of computationally useful formal languages, such as grammar and lexicon formalisms, for encoding linguistic knowledge.
3. Investigation of appropriate software architectures for various NLP tasks
4. Consideration of the types of non-linguistic knowledge that are an intrinsic part of NLP.

"It is an abstract area of study and it is not one that makes particular commitments to the study of the human mind, nor indeed does it make particular commitments to producing useful artifacts." [33]

LE is the application of NLP for the construction of computer systems that process language for some task usually other than modeling language itself, or "the instrumental use of language processing, typically as part of a larger system with some practical goal, such as accessing a database"[34].

Cunningham in Chapter 2 of his thesis on Software Architecture for Language Engineering [17] provides a good overview of the history of Language Engineering and broader definitions of NLP.

5.3.2. Language Engineering Transducers / Modules

GATE is compliant with the TIPSTER architecture [32] and is freely available, including source. The system consists of *transducers* or modules that at each step add structure, and often lose information, hopefully irrelevant [32].

Summarized from the TIPSTER architecture a minimum of at least one of the following transducers are required for the building of an Information Extraction (IE) system.

1. **Text Zoner -** turns a text into a set of text segments, minimally it would separate the formatted from the unformatted region.
2. **Preprocessor -** turns a text or text segment into a sequence of sentences, each of which is a sequence of lexical items. A lexical item is a word together with its lexical attributes. This module minimally determines the possible parts of speech for each word, and may choose a single part of speech. A good example of the a lexical engine is WordNet [46]
3. **Filter -** turns a set of sentences into a smaller set of sentences by filtering out the irrelevant ones. Level of relevancy may be achieved by detection of keywords in the sentence.
4. **Preparser -** takes a sequence of lexical items and tries to identify various reliably determinable, small-scale structures.
5. **Parser -** whose input is a sequence of lexical items and perhaps small-scale structures (phrases) and whose output is a set of parse tree fragments, possibly complete.
6. **Fragment Combiner -** tries to turn a set of parse tree or logical form fragments into a parse tree or logical form for the whole sentence.
7. **Semantic Interpreter** - generates a semantic structure or logical form from a parse tree or from parse tree fragments.
8. **Lexical Disambiguation -** turns a semantic structure with general or ambiguous predicates into a semantic structure with specific, unambiguous predicates.

9. **Coreference Resolution or Discourse Processing -** turns a tree-like structure into a network-like structure by identifying different descriptions of the same entity in different parts of the text.
10. **Template Generator -** derives the templates from the semantic structures.

5.4. GATE Components and Architecture

To use and understand GATE the knowledge of the following terminology is essential. The use of the API and understanding the documentation becomes simpler if these concepts are kept in mind:

- **Language Resource (LR):** refers to data-only components such as lexicons, corpora, thesauri and ontologies.
- **Processing Resource (PR):** refers to components whose character is principally task oriented or algorithmic, such as lemmatisers, generators, translators, parsers and parts of speech recognizers.

The basic components of GATE comprises of the following 3 principal elements as illustrated in figure 5.7:

1. **GDM -** the GATE Document Manager, based on the TIPSTER document manager. It provides a central repository that stores all the information an LE system generates about the text it processes. The core of the model are the documents, which are grouped into *collections* (or corpora) containing text and the annotations upon them.

2. **CREOLE -** a Collection of REusable Objects for Language Engineering. Usually a wrapper around a pre-existing LE component integrated with the system. Upon the initiation of a particular CREOLE via the GGI, or programmatically via the GATE API, the object obtains the necessary information (the documentation source and annotations created by other PRs) via calls to the GDM API.
3. **GGI -** the GATE Graphical Interface, a development tool for LE Research and Development, provides integrated access to the services of the other components and adding visualization and debugging tools. The GGI is also provided as a service at the University of Sheffield website.

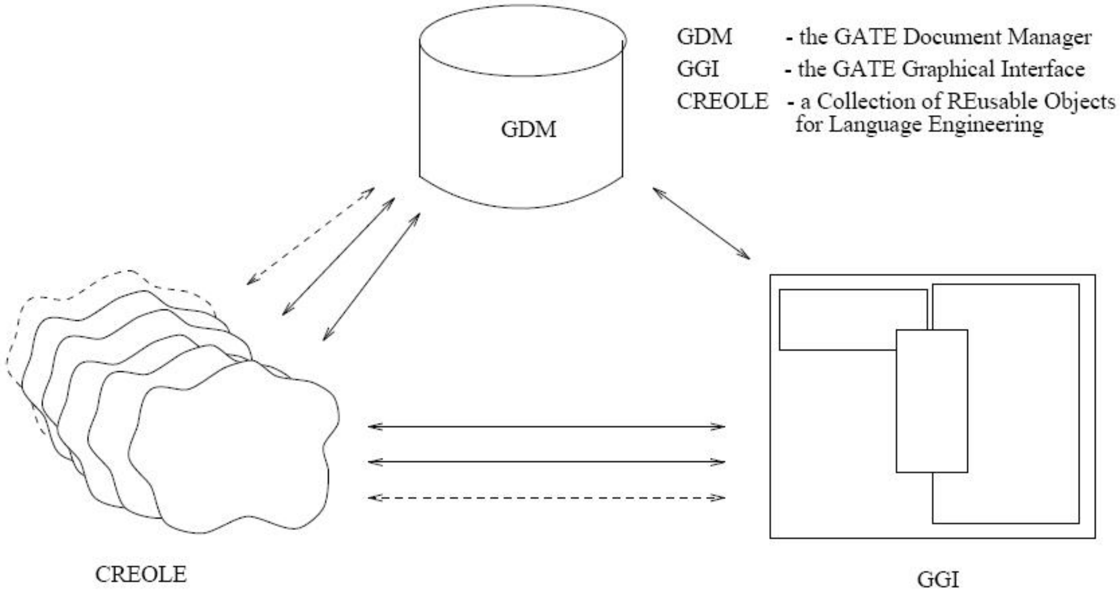

GDM
GDM - the GATE Document Manager
GGI - the GATE Graphical Interface
CREOLE - a Collection of REusable Objects for Language Engineering
CREOLE
GGI

Figure 5.7 GATE components

5.5. Putting it all together - ANNIE

A Nearly-New Information Extraction System (ANNIE) is the Information Extraction tool that comes with GATE. It's robust and open-source features played a significant role in it's integration with SEES. It uses the GATE architecture and the following PRs:

1. **Tokeniser** - splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types (e.g. with an initial capital, all upper case, etc.). The aim is to limit the work of the tokeniser to maximize efficiency, and enable greater flexibility by placing the burden of analysis on subsequent tools.
2. **Sentence Splitter** - segments the text into sentences. This module is required for the tagger. Both the splitter and tagger are domain and application independent.
3. **Tagger** - a modified version of the Brill tagger, which produces a part-of-speech tag as an annotation on each word or symbol. Neither the splitter nor the tagger is a mandatory part of the IE system, but the extra linguistic information they produce increases the power and accuracy of the IE tools.
4. **Named Entity Recognizer** - consists of pattern-action rules, executed by the finite-state transduction mechanism. It recognizes entities like person names, organizations, locations, money amounts, dates, percentages, and some types of addresses.

Figure 5.8 shows a document's route via ANNIE's modules defined above, in context with the GATE architecture and LE concepts describe in the previous sections and Information Extraction concepts in Chapter 4. The figure also shows components of LaSIE (Large Scale information Extraction), another project base on the GATE architecture at the University of Sheffield NLP group.

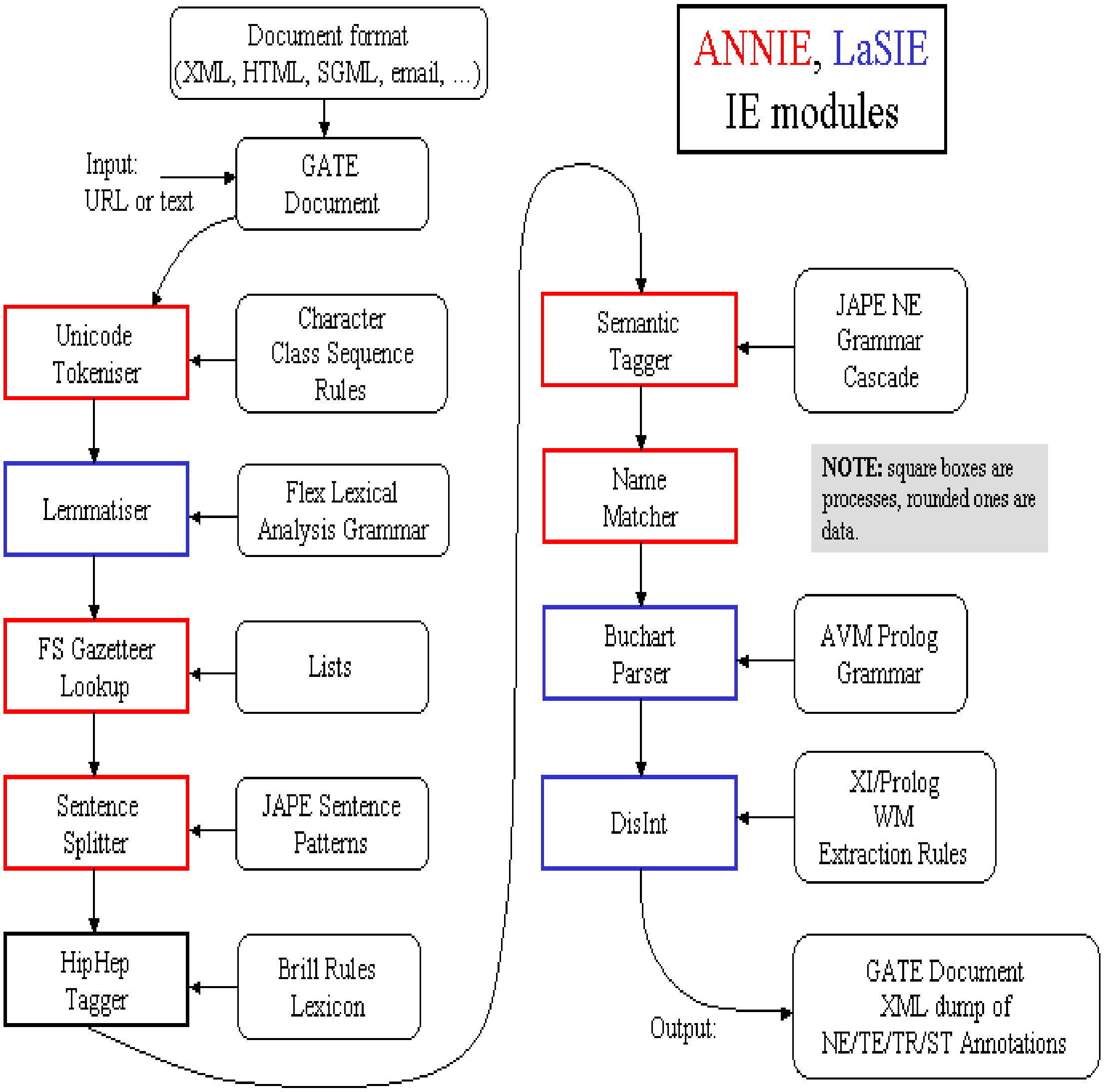


Figure 5.8 ANNIE Data Flow [37]

CHAPTER 6

SEES IMPLEMENTATION

SEES is a Mutimodal, Adaptive User Interface that provides a speech and telephony interface to POP3 email and RSS feeds. Using ANNIE, the system populates a relational database with the contextual information present in the subscribed emails and RSS feeds. This contextual information provides the basis for the novel approach taken to form the command-and-control mechanism for the Speech Recognition Engine.

The Mutimodal user interface allows users to access the email or RSS feeds via both the classical WIMP and speech interface. The unique filtering feature provided by the client serves as a powerful tool for content-filtering and boosts the Information Extraction process.

Functions to delete and traverse through the emails have been implemented, and work using both voice recognition or user inputs via the GUI. Speech Recognition and Text-to-Speech for the application were built into the system using Microsoft's Speech Application Protocol Interface (SAPI), using the standard SR and TTS Engines provided by Microsoft with their Speech add-in for Windows XP. Windows TAPI was used to enable the application to seamlessly work over a telephony line.

6.1. Background

Multi-modal interfaces are being developed so that one is not bound to interacting using a single type of input or output method. These interfaces though very useful on their own accord, if extended with built in knowledge can come close to accomplishing what Hankock and Chignell defined as an “intelligent interface” which “must mediate between two or more interacting agents who possess an incomplete understanding of each others’ knowledge and or form of communication” [1].

SEES is such a system. It manipulates electronic documents and categorizes the contents according to semantic criteria – dates, names, locations etc. This knowledge base is used for building the grammar for the speech interface, which provides a more natural command and control language and a novel filtering system for the content.

The interface proposed is multi-modal, a Graphical User Interface with all commands having voice recognition capability. SEES is also fully accessible via a telephone line and can be considered ubiquitous [6].

The system will not only benefit the disabled community, but also anyone in need of accessing information quickly and efficiently, not being hindered by modes of input or output, or for requiring full-text searches.

The need to develop such a system arose from the University of Texas at Arlington (UTA) Computer Science Engineering team that has developed a hand-held solution to provide assistance to the disabled. These researchers have developed wireless communication devices known as Personal Portable Devices, or PPDs that are customized for ease of use based on a person's disability and programmed to meet an individual's specific needs and interests [7].

The intention for these PPDs is to serve not only as a two-way communication device but also to get real-time messages – lending itself to health care monitoring.

Enabling easy access to the World Wide Web and email services on theses devices, particularly for the blind or visually impaired, would further enhance the usefulness of these PPDs.

6.2. Context Retrieval and Storage

As seen in Chapter 3 there are a number of applications that have attempted to provide a speech enabled front-end to an email client. However, these systems do not provide any methods of retrieving or storing the contextual information that these emails, and in our case, RSS feeds may possess. The approach of these interfaces is to make the simple parts of an email client speech enabled and provide basic functionalities such as ‘Compose Mail’, ‘Read Mail’ via a desktop/web client or through a telephony interface.

Searches for references and filters provided for the email client are either not present or require the manual manipulation of lists or other settings present in the client. This in itself proves to be a task that may is cumbersome for the blind or visually impaired.

SEES presents a novel parsing and storage solution for the contextual information from the emails and RSS feeds using ANNIE as the LE tool. This information provides the building blocks for the dynamic grammar used for the processing of the utterances by the user.

6.3. Novel Navigation and Filtering System

For an HCI that interfaces to an email system using Speech Recognition and Text to Speech technologies, it would be rather cumbersome for the user if all the application did was to present the entire Inbox to the user via a line by line read-through.

Without the visual representation of the Inbox, it is hard to form a mental representation of the summary of the emails present. Powerful search, filtering and traversal functions have to be considered for any speech-based interface that tries to serve up this data without the aid of sight.

The search and traversal language that the system recognizes should be conversational, rather than formal. The interface should have queries that provide information about the state that the system is in. These queries should return information about the base result set and the filters that have been applied to the base result set.

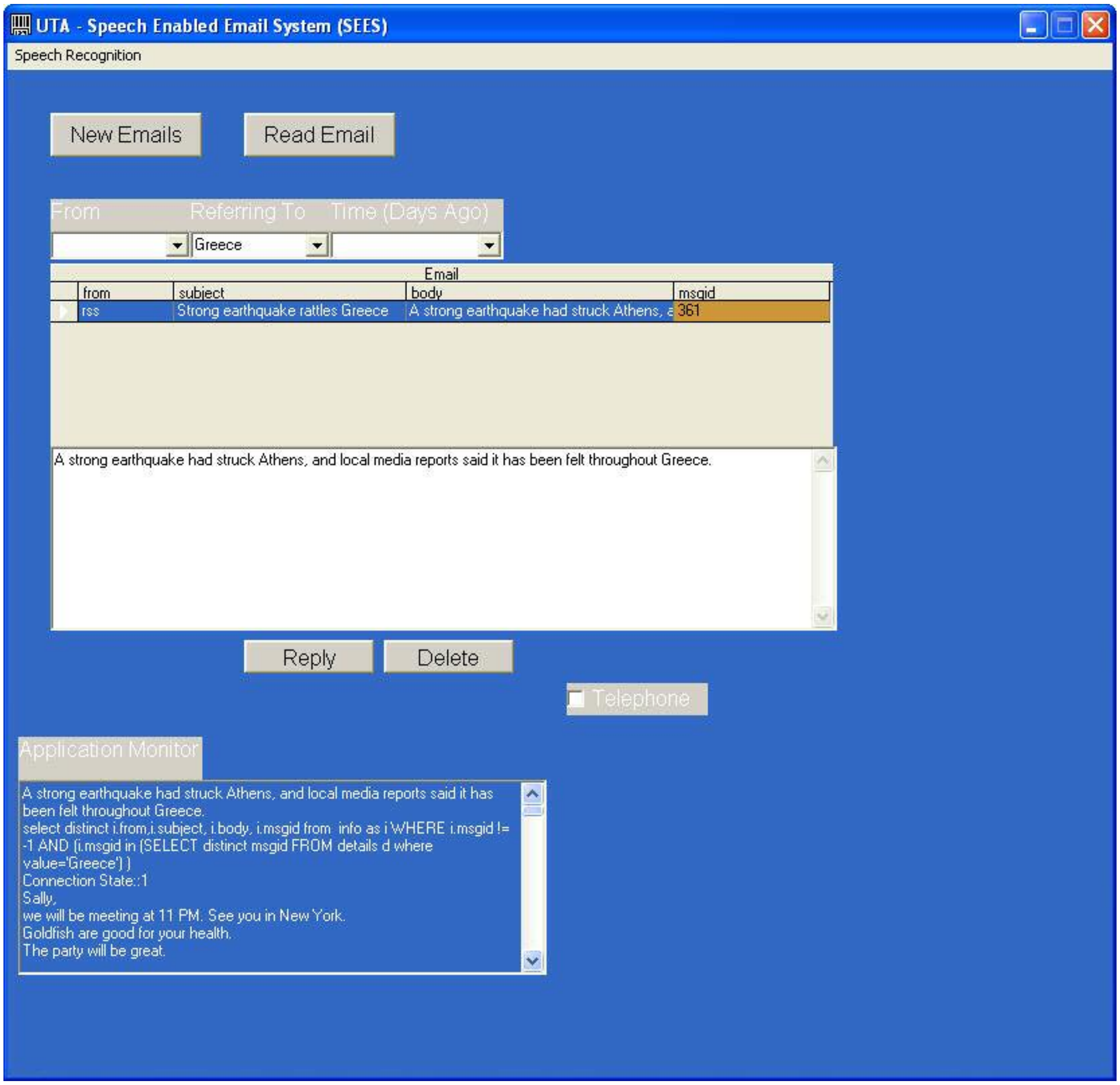
The interface presented retrieves the information for the user applying a novel approach that uses successive and cumulative filtering of the base result set. The base email set as the name indicates becomes the set of emails that have to be retrieved in order for the system to progress to any successive filtering states. The base result set can be retrieved using any of the commands represented in the following methodologies:

1. **Vanilla Retrieval** – "New Emails", "Old Emails" or "All Emails"
2. **Referential Retrieval** – "New Emails from ***person"***, "Old Emails from ***person"*** or "All Emails from ***person"***

Upon the retrieval of the base set of emails, the system starts traversing through the list. The system is now in ready to accept either one or both of the following filters from the user:

1. **Context Based –** references to events, people, places dates and times within the Subject and Body of the email e.g. “referring to Texas”, “referring to Arthur” etc.
2. **Time Based** - takes into account the timestamp of the email, and can be referenced by days relative from today e.g. “from yesterday”, “ from 2 days ago” etc.

If successive filtering is not desired or a filtering state already known, the user can directly go to that state by providing a retrieval methodology and/or the filter(s) all at once. Figure 6.2 represents the transition from the base email retrieval to the filtering states.



From:
Referring To: Greece
Time (Days Ago):

Email

from	subject	body	msgid
rss	Strong earthquake rattles Greece	A strong earthquake had struck Athens, a	361

A strong earthquake had struck Athens, and local media reports said it has been felt throughout Greece.

```
A strong earthquake had struck Athens, and local media reports said it has
been felt throughout Greece.
select distinct i.from,i.subject, i.body, i.msgid from  info as i WHERE i.msgid !=
-1 AND (i.msgid in (SELECT distinct msgid FROM details d where
value='Greece') )
Connection State::1
Sally,
we will be meeting at 11 PM. See you in New York.
Goldfish are good for your health.
The party will be great.
```

Figure 6.1 Applying Filters in SEES

At any point of the traversal of the emails, a command can be given to read the email that the sender and the subject are being presented for. Visually a cursor represents the current email of the traversal function. The user also has the option of initiating a brand new query for the retrieval of the base email set at any point during his/her interaction with the system.

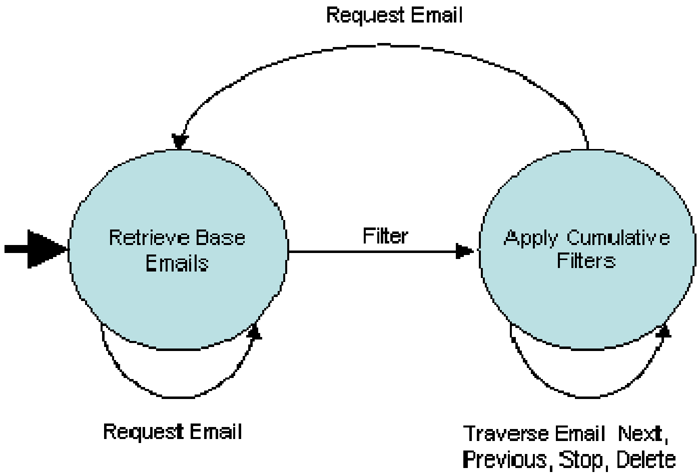


Figure 6.2 Basic Retrieve and Filter States of SEES

Once an email is chosen to be read by the user, the system accepts commands that are unique to this state. Positioning commands, that start the text-to-speech process from various parts of the body of the email, are provided. The user can forward or reply to the message once this state is reached.

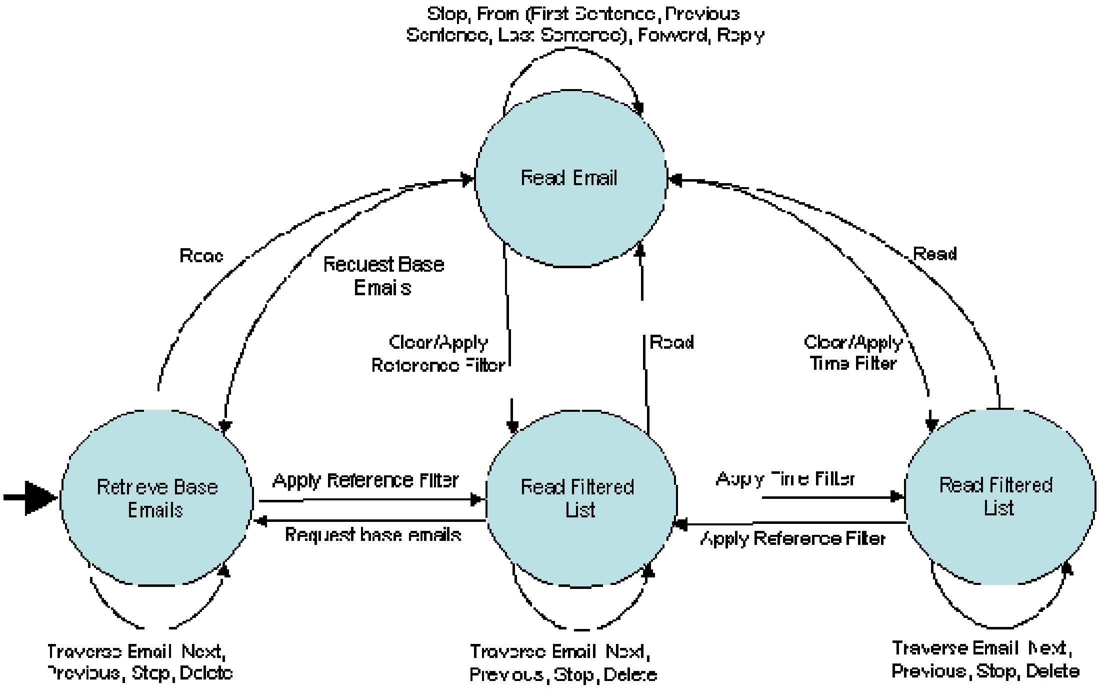


Figure 6.3 **Detailed States of SEES**

As an example:

"All emails from Bob" (Base Set)

Type	**Sender**	**Subject**	**Body**	**Days old**
Old	Bob	Party.	Are you bringing Sally?	1
New	Bob	venue.	Changed the venue to Central Park.	0
Old	Bob	Hi.	Party at my place - February 10th	7

All Filters are now applied to the Base Set.

"Referring to Sally" (also can be uttered as "All emails from Bob **Referring to Sally**")

Type	**Sender**	**Subject**	**Body**	**Days old**
Old	Bob	Party.	Are you bringing Sally?	1

"Referring to party" (also can be uttered as "All emails from Bob **Referring to party**")

Type	**Sender**	**Subject**	**Body**	**Days old**
Old	Bob	Party.	Are you bringing Sally?	1
Old	Bob	Hi.	Party at my place - February 10th	7

"From yesterday" (also can be uttered as "All emails from Bob **From yesterday**"

Type	**Sender**	**Subject**	**Body**	**Days old**
Old	Bob	Party.	Are you bringing Sally?	1

"Referring to Party and February" (also can be uttered as "All emails from Bob **Referring to party and February**").

Type	**Sender**	**Subject**	**Body**	**Days old**
Old	Bob	Party.	Are you bringing Sally?	1
Old	Bob	Hi.	Party at my place - February 10th	7

6.4. Use Cases

The following use cases were used as a skeleton to develop SEES.

1. **Retrieve new/old/all emails from Joe.**

   a) Using the microphone say "Retrieve new/old/all emails from Joe".

   Or

   b) Click on the GUI and select "Joe" from the "Retrieve new/old/all emails from:" drop list. The system displays the emails the subjects of all new/old/all emails from Joe on the screen, and, if multiple, reads the first subject. The user can then say or click "Read Email" to read the email, or say or click "Next email" to traverse to the next message. At any point the command "Stop" can be spoken to halt any text-to-speech activity being performed by the client

   Or

   c) Dial the telephone to a predefined number computer and follow the same procedure as a).

2. **Retrieve new/old/all emails.**

   Similar to 1.

3. Retrieve new/old/all emails from Mom.

Similar to 1.

4. Retrieve new/old/all emails referring to Jack.

Similar to 1.

5. Retrieve new/old/all emails referring to Texas.

Similar to 1.

6. Retrieve new/old/all emails referring to job postings.

Similar to 1.

7. Retrieve new/old/all emails from Mom referring to Joe from yesterday

Similar to 1.

8. Compose an email to James.

a) Using the microphone say "New email to James". The system prompts you for James' address if it doesn't find it in your address book. It then prompts you for a subject line – upon completion you say "done subject". It then prompts you to dictate the email. Once dictation is complete you say "send email now".

Or

b) Click on the GUI and select compose email. Type the email address, subject and body of email and click send.

Or

c) Dial the telephone to a predefined number and follow the same procedure as a).

9. **Retrieve today's headlines from USA today.**

    a) Using the microphone say "Headlines from USA today".

    The system displays and starts reading the news headlines from USA today.

    Or

    b) Click on the GUI and select the Read Headlines from USA today button. The system displays and starts reading the headlines from USA today.

    c) Dial the telephone to a predefined number and follow the same procedure as a).

10. **Search the web and retrieve the first document with references to the Tsunami relief effort that is returned by the search engine Google**.

    a) Using the microphone say "Retrieve from web Tsunami Relief Effort". The system displays the web page retrieved and reads it.

    b) Click on the GUI and type in "Tsunami relief effort" on the query input box that says "Retrieve from web" and click on the "Retrieve from web" button. The system displays the web page retrieved and reads it.

    c) Dial the telephone to a pre-defined number and follow the same procedure as a).

6.5. Architecture

SEES comprises of three major components. They are:

1. **Monitor –** Extraction and processing of content from source.
2. **Data Store** – information storage by Monitor and retrieval by the clients.

3. **Client** – access to the processed information via telephony, voice or WIMP client.

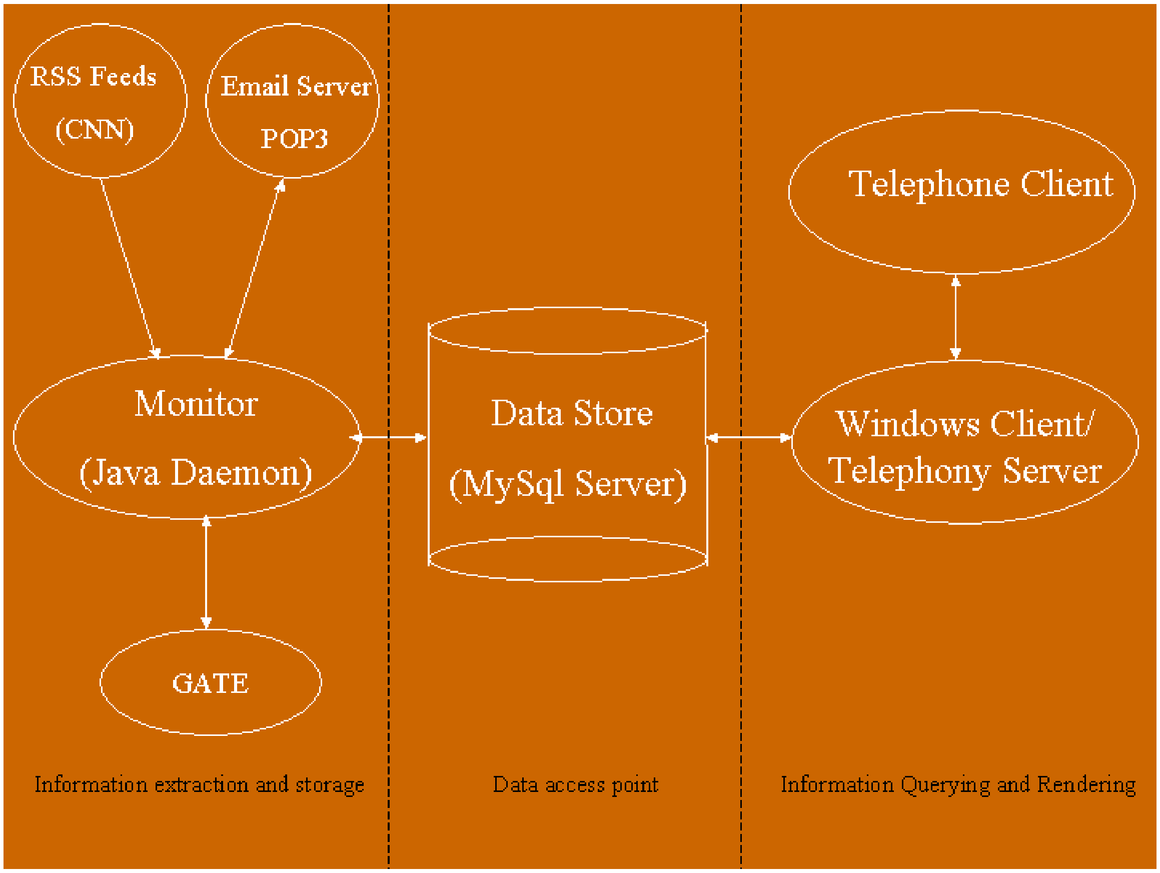
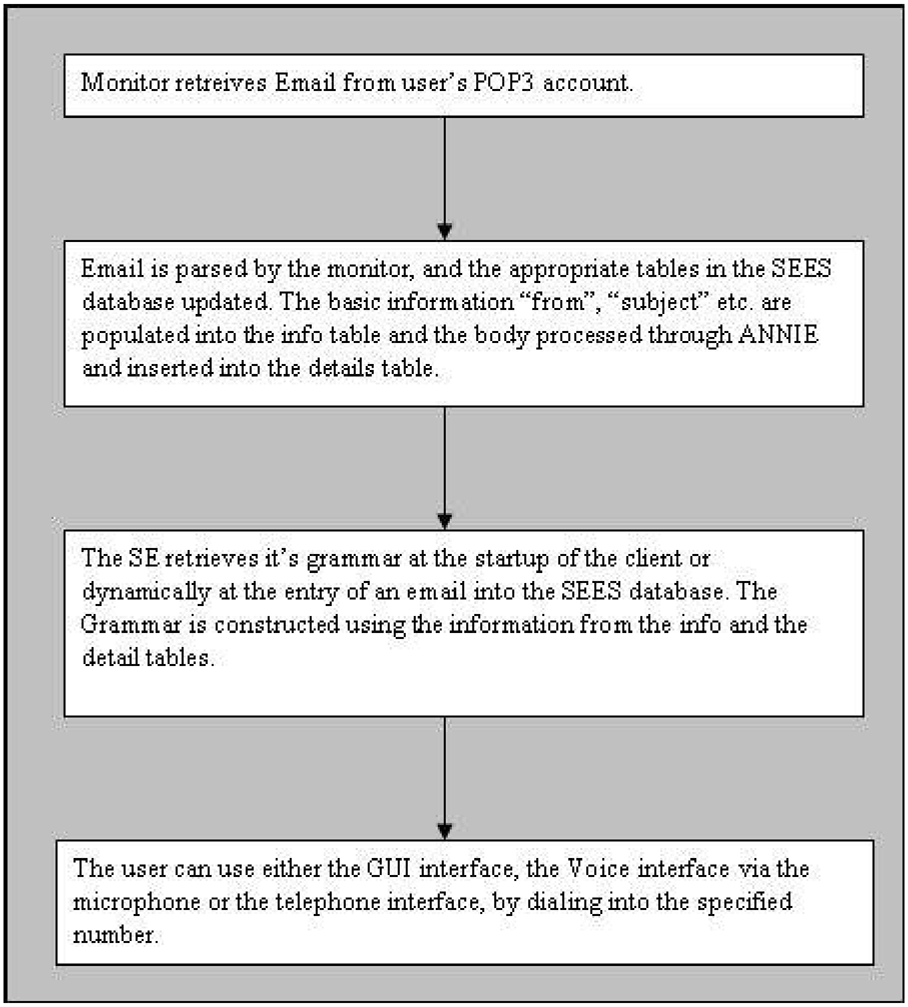


Figure 6.4 SEES Architecture

From figure 6.4 we see that this is not a standard client-server model. The database acts the middle-tier in this architecture. The client can still operate if the Monitor is not functioning, giving the user an off-line option i-e. not connected to the web while interfacing with the already processed emails that reside in the database.

The Data Store is the crucial component of the architecture. It's unavailability halts the functions of both the client and monitor.



Monitor retreives Email from user's POP3 account.
Email is parsed by the monitor, and the appropriate tables in the SEES database updated. The basic information "from", "subject" etc. are populated into the info table and the body processed through ANNIE and inserted into the details table.
The SE retrieves it's grammar at the startup of the client or dynamically at the entry of an email into the SEES database. The Grammar is constructed using the information from the info and the detail tables.
The user can use either the GUI interface, the Voice interface via the microphone or the telephone interface, by dialing into the specified number.

Figure 6.5 SEES Information and Process Flow

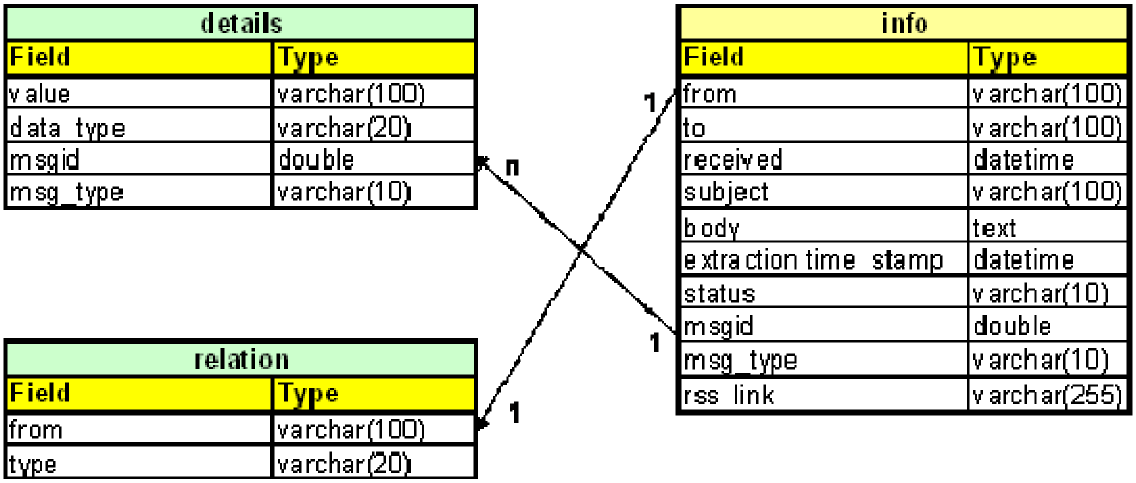
Each of the components is discussed in detail in the following subsections.

6.5.1. Data Store

The Data Store is a MySql database with a schema to accommodate the original and parsed emails and RSS feeds. There are 3 tables that the schema comprises of:

- **info** – This table is the master table. The emails and RSS links in their entirety are retrieved by the Monitor and placed in this table.

- **details** – Once ANNIE finishes with the Information Extraction process the details records corresponding to the email get populated.
- **relation** – The lookup table that the queries use for queries such as "New emails from ***Mom***". The hard coded emails with relationships or aliases are kept in this table.



details

Field	Type
value	varchar(100)
data type	varchar(20)
msgid	double
msg_type	varchar(10)

relation

Field	Type
from	varchar(100)
type	varchar(20)

info

Field	Type
from	varchar(100)
to	varchar(100)
received	datetime
subject	varchar(100)
body	text
extraction time stamp	datetime
status	varchar(10)
msgid	double
msg_type	varchar(10)
rss link	varchar(255)

Figure 6.6 SEES schema

6.5.2. Monitor

The Monitor is a daemon that establishes a connection with specified email server(s) and monitors it for new emails, extracts pertinent information from it and stores it in a relational database. The monitor also subscribes to RSS feeds and performs the function of processing and storing data from them. The Monitor is written in Java. It uses the standard JavaMail API available from Sun [41] and the built in java.net.url library to retrieve the emails and RSS feeds respectively.

The GATE API for ANNIE is used to process text, as discussed in Chapter 5. The *corpora* in our case become the emails and the RSS feeds.

Credentials for the email addresses being monitored, such as user name, password and the POP3 server, are provided to the daemon in a configuration file. The frequency for checking mail is also specified in this file. This file also contains the RSS feeds to be monitored.

Once the daemon establishes a connection with the POP3 server it provides the appropriate credentials of the user and proceeds to retrieve all new emails from the Inbox. The retrieved emails are stored in the SEES database info table and marked 'New' in the status field.

The monitor then invokes ANNIE to extract the lexical information from the Body and Subject of these emails. This list of lexical items associated with the email is stored in the details table. The monitor then marks these emails as 'complete' in the database.

6.5.3. SEES Client

The client is the content delivery component of SEES. It also doubles as the telephony server, enabling the use of the client via a telephone. The client is written in Visual Basic and connects to the SEES database via ODBC. The Microsoft Speech SDK 5.1 is used to make the client multimodal by providing all the WIMP functionality via voice.

SEES can operate in three different client modes. They are:

1. **Desktop Client with speech –** multimodal, use of mouse and keyboard available
2. **Desktop Client without speech –** only mouse and keyboard.
3. **Telephony client –** interaction via a telephone, using speech.

The Desktop Client with and without speech has a straightforward setup. By default the application starts up with in Desktop Client with speech mode. In order to switch between these two modes one can use the shortcut keys Ctrl+E or Ctr+S or use the menu bar as shown in figure 6.8.

In order to access the system via the telephone the user dials in to the telephone number that the windows based client is attached to. The client application is set to "Telephone Mode" by:

a) uttering the phrase "Phone Mode" , if in speech mode

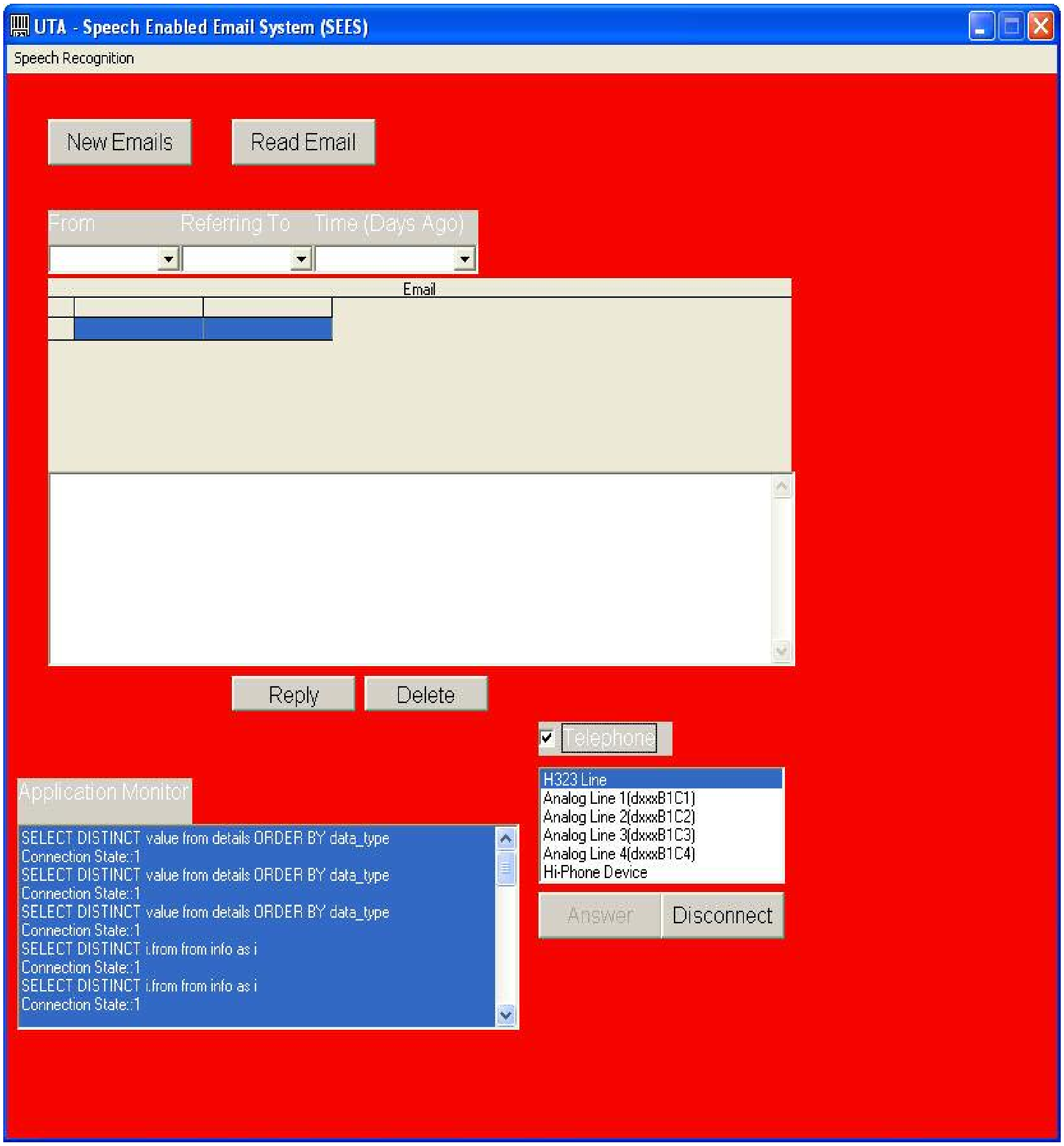
b) clicking the "Telephone" checkbox

c) using the Ctrl+T shortcut to enable the checkbox (figure 6.8)

Once any of the above events occur the client is set to answer incoming calls, provided a TAPI compliant modem/telephony board is present. Specifications for the modem are mentioned in Chapter 5.

Microsoft TAPI exposes the features to detect, answer and communicate with the incoming call to the Visual Basic client, thereby enabling it to handle the phone-line communications. The client also has the ability to receive communications using the H232 protocol, which handles VOIP as described in Chapter 5.

The user dials into the number that the client is attached to. Upon the detection of an incoming call the client answers the call and presents the Welcome message. The client is then ready to process a voice response from the caller.



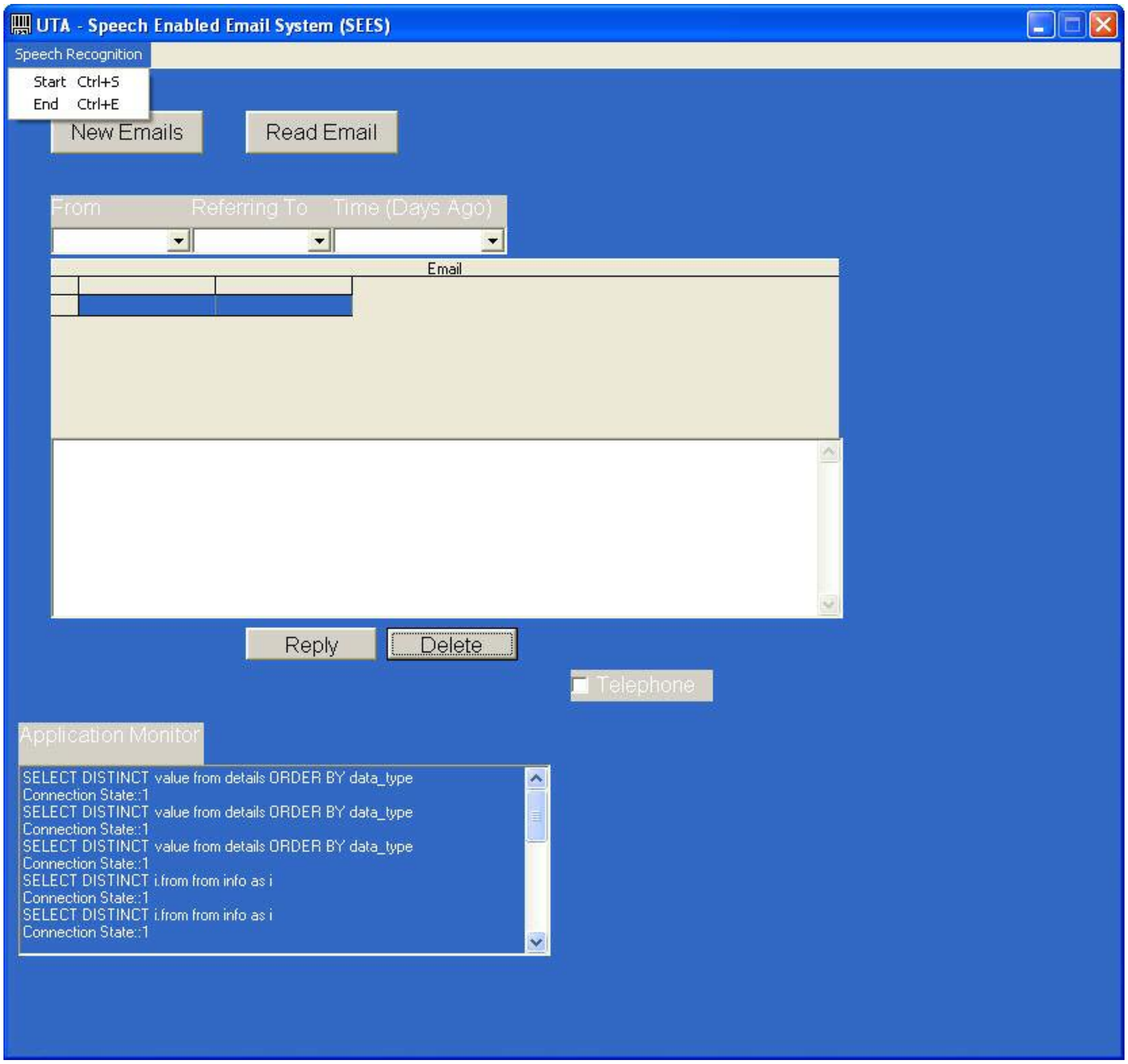
```
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT i.from from info as i
Connection State::1
SELECT DISTINCT i.from from info as i
Connection State::1
```

Figure 6.7 SEES Client Telephone Mode

From there on the interaction is identical to that of the voice driven windows client.

6.5.3.1.Client Speech Recognition

The client uses the Speech Engine's command and control methods to process any speech inputs, to limit any ambiguity of the speaker's intent. The commands are predefined, but the grammar for these commands is dynamically updated upon the extraction of new content from the emails or RSS feeds.



```
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT value from details ORDER BY data_type
Connection State::1
SELECT DISTINCT i.from from info as i
Connection State::1
SELECT DISTINCT i.from from info as i
Connection State::1
```

Figure 6.8 Enabling Speech Recognition

The user can interchangeably use the speech commands or perform the available functions using the GUI. All functions that are represented on the GUI are achievable by using the voice mode and vice-versa.

When the client is invoked it assesses the new emails and RSS feeds that have been populated in the database. The grammar is built using the information extracted by

ANNIE. When the client starts up it initializes it's grammar by extracting the information required for the command and control grammar.

As an example, for use case #1, the phrase "New emails from" is recognized through a series of procedures that involve all components of the SEES architecture (figure 6.3).When a new email is extracted and it's information from ANNIE updated in the database, the client expands it's grammar dynamically. The name of the sender is stripped of his address and any special characters in his email id e-g. John<John.Doe@jondo.com> is translated to "John".

The name is then added at the end of the phrase and the SRE's grammar reinitiated with the new phrase "New emails from John". The client then has the ability to recognize the sentence via the voice interface which is controlled by SAPI. The name is also added to the ends of the phrases "Old emails from" and "All emails from", and the resulting phrases added to the grammar. The grammar is then updated with these phrases.

The SE in now ready to recognize the above three phrases and upon the utterance and successful recognition by the SE of any one the phrases, the client transforms the string into an SQL statement which reads as:

SELECT * FROM info

WHERE

info.status = 'new'

AND

info.from LIKE 'John%'.

Similarly for the phrase “New emails from Mom”, the Monitor makes the association of the email address that is tagged as “Mom” in the database and the Grammar is maintained in a static fashion for this phrase. Upon a successful recognition of the phrase the SQL transformed from this string would read as

```
SELECT * FROM info, relation
WHERE
info.from = relation.from
AND
relation.type=’MOM’
```

For the phrase “New emails referring to Texas”, the client uses the knowledge of the semantic information extracted by ANNIE and stored in the detail table of the SEES database. The annotated text “Texas” is successfully identified by the Monitor using ANNIE. The client forms the grammar by appending the Named Entities that ANNIE extracts to the phrase “New emails referring to “.

This same process is applied to “Old” and “All” emails referring to Texas. The corresponding SQL for a successful recognition of the phrase then translates into

```
SELECT * FROM info, detail
WHERE
info.msgid =* info.detail
AND
info.value = ‘TEXAS’
```

The other phrases work in a similar fashion.

6.5.3.2.Client Speech Synthesis

Speech synthesis occurs as a response for every command that is uttered and recognized during the interaction with the client in telephony or desktop speech interface mode. As an example the response to the query “New emails from John” would initiate a response “There are no new emails from John”, if there were no new emails received from John.

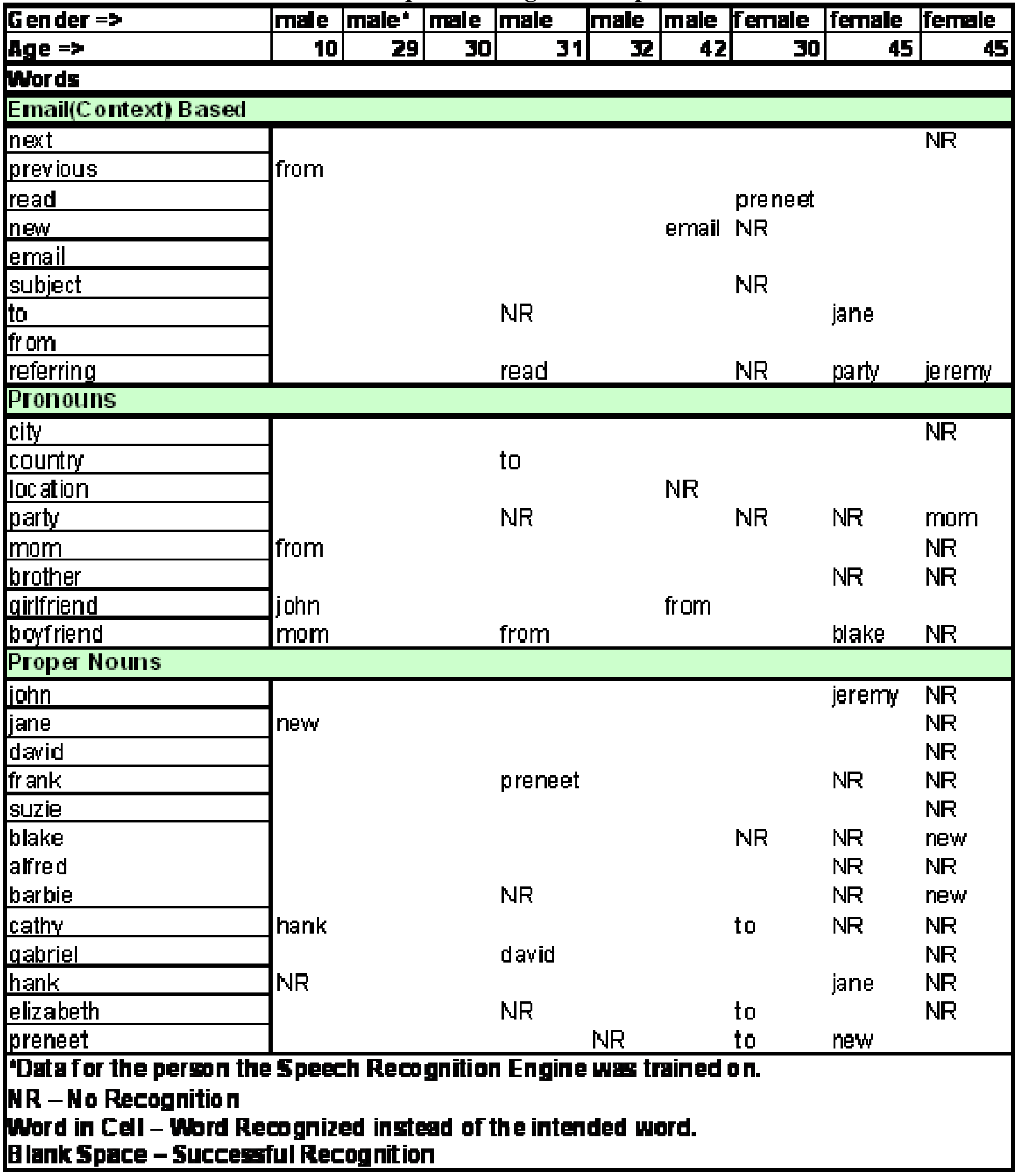
A welcome message “This is SEES”, is synthesized at the launch of the client or when a user calls in and the telephony client answers the call. TTS is achieved using Microsoft SAPI. An administrator can change the voice or the TTS engine as long as it is compatible with SAPI.

6.6. Speech Recognition Experiment

Since the system will use the Command and Control mode of the Speech Recognition Engine, experiments were conducted to see the feasibility of using the system as a client for multiple users, without having the need to train the SR Engine for each individual and then storing their profile.

Results of the experiment conducted are shown in Table 6.1. The results showed best results for the person who had trained the SR Engine and for persons similar in age and of the same sex as the trainer. The result was as expected, since the Microsoft SR Engine used was speaker-dependent and results are greatly affected by the amount of training provided by the user.

Table 6.1 Speech Recognition Experiment



Gender =>	male	male*	male	male	male	male	female	female	female
Age =>	10	29	30	31	32	42	30	45	45
Words									
Email(Context) Based									
next									NR
previous	from								
read							preneet		
new						email	NR		
email									
subject							NR		
to				NR				jane	
from									
referring				read			NR	party	jeremy
Pronouns									
city									NR
country				to					
location						NR			
party				NR			NR	NR	mom
mom	from								NR
brother								NR	NR
girlfriend	john					from			
boyfriend	mom			from				blake	NR
Proper Nouns									
john								jeremy	NR
jane	new								NR
david									NR
frank				preneet				NR	NR
suzie									NR
blake							NR	NR	new
alfred								NR	NR
barbie				NR				NR	new
cathy	hank						to	NR	NR
gabriel				david					NR
hank	NR							jane	NR
elizabeth				NR			to		NR
preneet					NR		to	new	

*Data for the person the Speech Recognition Engine was trained on.
NR – No Recognition
Word in Cell – Word Recognized instead of the intended word.
Blank Space – Successful Recognition

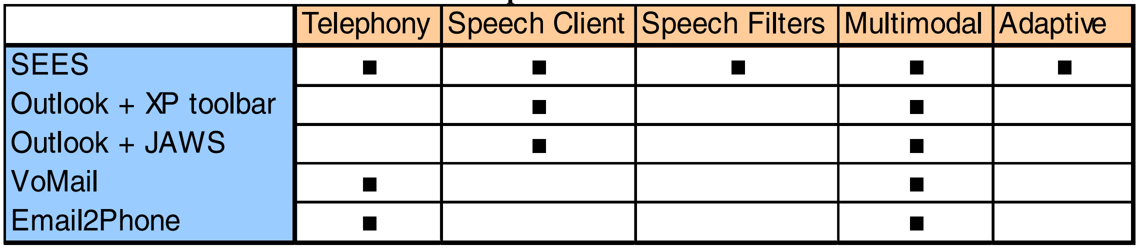
CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. Conclusion

“There is no Moore’s Law for user interfaces” [22]. The way we have interacted with computers has not changed much in the past few decades. WIMP interfaces still dominate the way we communicate with computers. With ever new computing needs and ever shrinking devices this paradigm shows the lack of scalability to meet these needs.

Table 7.1 SEES Comparison with other Email clients

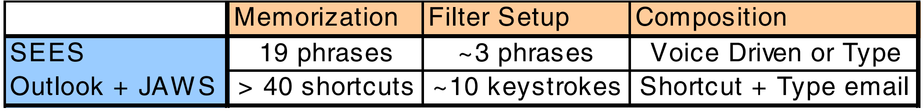


	Telephony	Speech Client	Speech Filters	Multimodal	Adaptive
SEES	■	■	■	■	■
Outlook + XP toolbar		■		■	
Outlook + JAWS		■		■	
VoMail	■			■	
Email2Phone	■			■	

As can be seen in table 7.1 there are no commercial applications that are addressing the needs for providing a suitable interface to leverage meaningful multimodal or adaptive speech interaction. Even though speech multimodality may exist it can only be used in conjunction with another mode, a definition that fits multimodality, however not suitable for a visually impaired person to perform any meaningful task using only the alternate mode for input.

Comparing SEES to the leading solution for email access for the blind – the combination of Microsoft Outlook and JAWS, the memorization of more than 40 keyboard shortcuts vs. the memorization of 19 key phrases, see Appendix A, without the need for filling any new information using the keyboard.

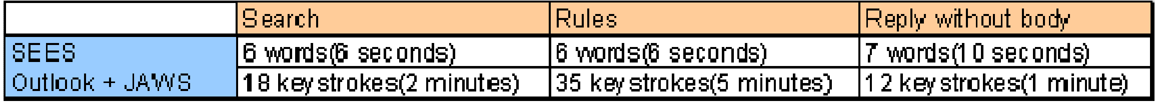
Table 7.2 SEES coparison with Outlook and JAWS



	Memorization	Filter Setup	Composition
SEES	19 phrases	~3 phrases	Voice Driven or Type
Outlook + JAWS	> 40 shortcuts	~10 keystrokes	Shortcut + Type email

The average time spent with JAWS reciting back the menus for a novice user is a hinderance and adds to the learning curve. In an experiment with 3 people who did not know either of the systems, not blind, however restricted to the use of the shortcuts in JAWS and voice commands in SEES respectively, it was found that the ease of use of SEES in finding emails and sending canned replies proved to be quick and efficient versus Outlook which requires more tedious data entry, as it is designed to serve as a broader group of people.

Table 7. 3 User Experiment SEES vs. Outlook and JAWS



	Search	Rules	Reply without body
SEES	6 words(6 seconds)	6 words(6 seconds)	7 words(10 seconds)
Outlook + JAWS	18 keystrokes(2 minutes)	35 keystrokes(5 minutes)	12 keystrokes(1 minute)

The experiment was based upon 10 emails. The search criteria was to find an email and reply to it with a canned reply of "received", "thanks" or "acknowledged".

1. From a certain person within a certain time frame
2. Referring to a certain city that the within the email within a certain time frame.
3. Referring to an event referred to within the email.

This small experiment shows that an application aimed at serving a specific user group with a disability rather than making a one size fits all application certainly helps in cutting down the learning curve as well as interaction times with the application.

During the course of this research it was a dismal finding that little is being done to take existing speech technologies and making use of them in innovative ways to help the disabled of today. There are currently research areas that will reap technology that will undoubtedly help the disabled, but as SEES shows the use of what we have today can be beneficial for the disabled, here and now.

7.2. Future Work

At present the entire SEES architecture is based upon a single user. The users must install all components on their local system in order to run SEES. This and other similar issues can be addresses by taking the following steps for making SEES a maintainable and scalable enterprise system:

7.2.1. Client

The client is process and memory intensive, or heavy. It plays the dual role of acting as a telephony server as well as the e-mail client. The client can only be configured for handling a single user.

In order to take away the hassle of running the client on a user's desktop, and occupying their phone line for checking mail remotely, part of the functionality of the client should be moved onto a client / server architecture. This can be achieved by:

a) Decoupling the component from the client that does the job of creating the Grammar for the Speech Recognition Engine, and develop it as a server that can handle multiple users.

b) Moving the WIMP and speech-enabled functionality of the client to a web-based application

c) Developing the telephony component such that it resides on a server and is not limited to a single line running on the user's machine.

7.3. Monitor and Data Store

The Monitor and Data Store are architected to handle a single user configuration for a single POP3 server, and reside on the user's machine. Both these components can be re-architected to reside on a server other than the user's machine. This can be achieved by:

a) Running the monitor and database on a server class machine that monitors multiple users on multiple POP3 servers.

b) The schema of the database would require changes to handle information gathered by the monitor to store it correctly by user account and server.

c) The monitor could potentially monitor Mail Servers than POP3.

Or by

a) Integrating the functionality of the Information Extraction component, achieved by ANNIE, into a POP3 or another email server. The schema of the server would need to be modified to achieve the extra functionality provided by the SEES database.

b) Developing clients that handle the modified schema of this email server.

APPENDIX A

SEES XML GRAMMAR FOR SPEECH RECOGNITION ENGINE

```
<!-- The grammar tag surrounds the entire CFG description The language of the
grammar is English-American ('409') -->
<GRAMMAR LANGID="409">
        <DEFINE>
                <ID NAME="closeapp" VAL="1" />
                <ID NAME="newemails" VAL="2" />
                <ID NAME="newemailsfrom" VAL="3" />
                <ID NAME="reademail" VAL="4" />
                <ID NAME="nextemail" VAL="5" />
                <ID NAME="previousemail" VAL="6" />
                <ID NAME="stopeverything" VAL="7" />
                <ID NAME="deleteemail" VAL="8" />
                <ID NAME="currentemail" VAL="9" />
                <ID NAME="traverseemails" VAL="10" />
                <ID NAME="newemailsreferringtoago" VAL="11" />
                <ID NAME="clearfrom" VAL="12" />
                <ID NAME="cleartime" VAL="13" />
                <ID NAME="clearreffering" VAL="14" />
                <ID NAME="clearquery" VAL="15" />
                <ID NAME="telephonemode" VAL="16" />
                <ID NAME="canceltelephonemode" VAL="17" />
                <ID NAME="replyemail" VAL="18" />
```

```
    </DEFINE>

    <RULE NAME="telephonemode" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>+phone mode</P>

    </RULE>

    <RULE NAME="canceltelephonemode" TOPLEVEL="ACTIVE"

EXPORT="1">

        <P>+cancel phone mode</P>

    </RULE>

    <RULE NAME="closeapp" TOPLEVEL="ACTIVE">

        <P>close application</P>

    </RULE>

    <RULE NAME="newemails" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>new emails</P>

    </RULE>

        <RULE NAME="reademail" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>read email</P>

    </RULE>

    <RULE NAME="nextemail" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>+next</P>

    </RULE>

    <RULE NAME="previousemail" TOPLEVEL="ACTIVE" EXPORT="1">
```

```
        <P>previous</P>
</RULE>
<RULE NAME="stopeverything" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>stop</P>
</RULE>
<RULE NAME="deleteemail" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>delete this email</P>
</RULE>
<RULE NAME="currentemail" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>current</P>
</RULE>
<RULE NAME="traverseemails" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>traverse</P>
</RULE>
<RULE NAME="cleartime" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>clear</P>
        <P>time</P>
        <P>filter</P>
</RULE>
<RULE NAME="clearfrom" TOPLEVEL="ACTIVE" EXPORT="1">
        <P>clear</P>
        <P>from</P>
        <P>filter</P>

    </RULE>

    <RULE NAME="clearreferring" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>clear</P>

        <P>referring</P>

        <P>filter</P>

    </RULE>

    <RULE NAME="clearquery" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>clear</P>

        <P>query</P>

    </RULE>

    <RULE NAME="replyemail" TOPLEVEL="ACTIVE" EXPORT="1">

        <P>reply saying</P>

        <L><P>yes</P>

        <P>no</P>

        <P>maybe</P>

        </L>

    </RULE>

</GRAMMAR>
```

REFERENCES

1. Chignell M.H. & Hancock P.A. Intelligent Interface Design. M. Helander (cd.) Handbook of Human-Computer Interaction, Elsevier, pp.969-995, 1988
2. Busby Dr G. TECHNOLOGY FOR THE DISABLED AND WHY IT MATTERS TO YOU. Computers in the Service of Mankind: Helping the Disabled (Digest No: 1997/117), IEEE Colloquium on
3. Buxton William. Human Interface Design and the Handicapped User. Conference on Human Factors in Computing Systems, (291) (6) (April 1986)
4. Marc Weiser. The Computer for the 21st Century. Scientific American, 265(3) (1991)
5. H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.
6. Mainstreaming the computer technology needs of disabled persons in higher education. Proceedings of the 17th annual ACM SIGUCCS conference on User Services, Maryland (75) (2) (1989)
7. Press Release, Tuttle Roger, October 2003, Homepage, http://www.uta.edu/engineering/press_releases/press_release.php?id=98

8. Gaze tracking for multimodal human-computer interaction Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on Volume: 4 , pp.21-24 April 1997
9. Oviatt Sharon, Ten Myths of multimodal interaction, , November 1999, Communications of the ACM
10. Langley Pat, User Modeling in Adaptive Interfaces, Proceedings of the Seventh International Conference on User Modeling. Banff, Alberta: Springer
11. Tsvi Kuflik, Peretz Shoval , Generation of user profiles for information filtering — research agenda (poster session), Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000
12. Lai Jennifer, Facilitating Mobile Communication with Multimodal Access to Email Messages on a Cell Phone, ,Conference on Human Factors in Computing Systems, Extended abstracts of the 2004 Conference on Human factors and Computing Systems, 2004, ACM
13. Handbook of Human-Computer Interaction, (ed. by J. Jacko & A. Sears), Lawrence Erlbaum: New/old/all Jersey, 2002.
14. Dino Esposito, The Microsoft Speech SDK, Microsoft Internet Developer, Feb. 1999 (http://www.microsoft.com/mind/0299/cutting/cutting0299.asp)
15. Microsoft Speech (http://www.microsoft.com/speech/evaluation/techover/)
16. Ubiquitous Computing(http://www.ubiq.com)
17. SALE thesis - ftp://ftp.dcs.shef.ac.uk/home/hamish/private/Thesis.pdf

18. SALE, http://gate.ac.uk/sale/tao/index.html#x1-1690008

19. Bolt, R. Put that there: Voice and gesture at the graphics interface. Computer Graphics (1980), 262-270.

20. Nigay, L. 1993. A Case Study of Software Architecture for Multimodal Interactive System: a voice-enabled graphic notebook, Technical Report CLIPS-IMAG, 31 pages.

21. Oviatt, S., Cohen, P. Multimodal interfaces that process what comes naturally. Comm. of the ACM, 43, 3 (2000), 45-53.

22. Matthew Turk, George Robertson, Perceptual user interfaces (introduction), Communications of the ACM, v.43 n.3, p.32-34, March 2000

23. Personal and ubiquitous computing [1617-4909] Bell yr:2007 vol:11 iss:2 pg:133

24. Bolt, R.A. Put that there: Voice and gesture at the graphics interface. ACM Computer Graphics 14, 3 (1980), 262–270.

25. TAPI, http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/cnet/cnad_arc_kxjc.mspx?mfr=true

26. Amundsen, Michael MAPI, SAPI, & TAPI Developer's Guide. SAMS, 1996

27. Microsoft Windows Net Meeting, http://www.microsoft.com/windows/netmeeting/

28. Dragon Naturally Speaking, http://www.nuance.com/naturallyspeaking/

29. SPHINX, http://cmusphinx.org/

30. Festival, http://www.speech.cs.cmu.edu/festival/

31. Eclipse, http://www.eclipse.org/

32. R. Grishman. TIPSTER Architecture Design Document Version 2.3. Technical report, DARPA, 1997.
http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/

33. G. Gazdar. Paradigm merger in natural language processing. In R. Milner and I. Wand, editors, Computing Tomorrow: Future Research Directions in ComputerScience, pages 88–109. Cambridge University Press, 1996.

34. Thompson H. Natural language processing: a critical analysis of the structure of the field, with some implications for parsing. In K. Sparck-Jones and Y. Wilks, editors, Automatic Natural Language Parsing. Ellis Horwood, Chichester, 1985.

35. Grishman R., Sundheim B. Message understanding conference - 6: A brief history. In Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, June 1996.

36. Gazdar G., Mellish C. Natural Language Processing in Prolog. Addison-Wesley, Reading, MA, 1989.

37. ANNIE, http://gate.ac.uk/sale/tao/index.html#annie

38. Email2phone, http://www.Email2phone.net

39. AdaptiveTechnologies, http://www.microsoft.com/enable/at/types.aspx

40. Chieko Asakawa, What's the web like if you can't see it?, Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A), May 10-10, 2005, Chiba, Japan

41. JavaMail API, http://java.sun.com/products/javamail/

42. Ainsworth W. Speech Recognition by Machine Peter Peregrinus / IEE, London, 1988.

43. Web Content Accessibility Guidelines 1.0, W3C Recommendation, May 1999, http://www.w3.org/TR/WAI-WEBCONTENT/

44. Wikipedia, The free Encyclopedia, http://en.wikipedia.org/

45. Schomaker, L., J.Nijtmans, A.Camurri, F.Lavagetto, P.Morasso, C.Benoit, T.Guiard-Marigny, B. Le Goff, J.Robert-Ribes, A.Adjioudani, I.Defee, S.Munch, K,Hartung, J.Blauert (1995) A Taxonomy of Multimodal Interaction in the Human Information Processing System, A Report of the Esprit BRA Project 8579 MIAMI, WP1, February 1995, http://www.nici.kun.nl/~miami/reports/reports.html

46. WordNet, http://wordnet.princeton.edu/

BIOGRAPHICAL INFORMATION

Aparajit Saigal received his Bachelor's from the University of Texas at Arlington in 1998. He joined National Semiconductor in 1997 as a co-op and has been with them since. He is currently a Staff Engineer with the company, involved with semiconductor fab automation, application development, data warehousing and simulations. Aparajit has been involved with his Master's program since 2001, while working and gaining industry experience at the same time. His research interests include Databases, Algorithms, Speech technologies and Language Engineering.