COMPARING SPEECH RECOGNITION AND TOUCH TONE AS INPUT MODALITIES FOR TECHNOLOGICALLY UNSOPHISTICATED USERS

By

Petrus Lineekela Kafidi

Submitted in partial fulfilment of the requirements of the degree
Master of Science (Computer Science)

in the

Faculty of Engineering, Building Environment and Information Technology

UNIVERSITY OF PRETORIA

June 2005

COMPARING SPEECH RECOGNITION AND TOUCH TONE AS INPUT MODALITIES FOR TECHNOLOGICALLY UNSOPHISTICATED USERS

Using an automated service to access information via the telephone has become an important productivity enhancer in the developed world. However, such automated services are generally quite inaccessible to users who have had little technological exposure. There has been a widespread belief that speech-recognition technology can be used to bridge this gap, but little objective evidence for this belief has been produced.

To address this situation, two interfaces, touchtone and speech-based, were designed and implemented as input modalities to a system that provides technologically unsophisticated users with access to an informational/transactional service. These interfaces were optimised and compared using transaction completion rates, time taken to complete tasks, error rates and user satisfaction. The speech-based interface was found to outperform the touchtone interface in terms of completion rate, error rate and user satisfaction. The data obtained on time taken to complete tasks could not be compared as the DTMF interface data were highly influenced by people who are not technologically unsophisticated. These results serve as a confirmation that speech-based interfaces are more effective and more satisfying and can therefore enhance information dissemination to people who are not well exposed to the technology.

KEYWORDS

Human-Computer Interaction, Speech Recognition, Usability, User Interface, Telephony Interface

1. INTRODUCTION

This chapter provides background information on human-computer interaction, and introduces the purpose of this thesis. The methodology used will also be described, as well as the high-level structure of the thesis.

1.1 BACKGROUND

The field of Human Computer Interaction is relatively new when compared to other fields of computer science such as Software Engineering. It has developed over a period of twenty years. John Karat of IBM sums up the development of HCI during the period 1980-2002 thus "…*from interface to interaction, from slow changes in technology to rapid changes, from a few users to everyone, from office work productivity focus to broad range of use.*" [Karat 03]. In the past twenty years the most visible change was from Command Line Interfaces to Graphical User Interfaces. However, progress has generally been confined to people who are well exposed to the technology. Thus, the information stored on computers has been available mainly to those who can access a computer via the mouse-keyboard interface. With the imminent successful deployment of telephony interfaces, that situation is about to change. With telephony interfaces, users can access information stored on a computer system using a standard telephone. Since telephones are highly accessible, making information available via the telephony interface will improve on the accessibility of information to technologically unsophisticated users.

Telephone interfaces can be implemented using Dual Tone Multi Frequency (DTMF or "touchtone") technology or as Speech Recognition interfaces. But which one will be most usable to technologically unsophisticated users? In other words, which of the two interfaces would be efficient, effective and satisfying when being used by such users? To answer these questions, examples of both interfaces were designed and implemented. Technologically unsophisticated users were then allowed to use the interfaces and thereafter the performance was evaluated and compared. This thesis describes the process and the outcome thereof.

1.2 OBJECTIVES

The main objective of this research project was to design, implement, evaluate and compare the performance of touchtone and speech based interfaces. In optimizing these interfaces, cultural and Human-Computer Interaction issues were studied and taken into consideration.

1.3 ORGANIZATION OF THE STUDY

Chapter 2 discusses Human Computer Interaction in general, emphasizing information exchange, processing and storage with respect to both humans and computers. Chapter 3 discusses the technologies underlying the two interfaces, i.e. speech recognition and DTMF processing. Chapter 4 describes the development and evaluation of the two interfaces. Chapter 5 presents the results obtained during the evaluation as well as discussions of these results. Chapter 6 puts forward the conclusions drawn from the research project and gives suggestions for future research in relation to this topic.

1.4 SCOPE AND LIMITATIONS

This thesis presents an overview of the HCI and the two technologies (DTMF and speech recognition) from an interface designer's perspective. It has put emphasis on the functionalities and characteristics of the technologies rather than the implementation issues. For detailed discussions on these topics, the reader is encouraged to follow the references as provided.

1.5 RESEARCH METHODOLOGY

Besides the references referred to, the Internet has been a valuable resource for obtaining information about the topic. Many of the materials referenced in this work were available on the Internet, and consequently in the reference section, Uniform Resource Locators (URLs) have been provided which were current at the time of research. However, due to the dynamic nature of the Internet and particularly the

World Wide Web, it cannot be guaranteed that this information remains available at these locations.

1.6 ACKNOWLEDGEMENT

I would like to thank my supervisor, Professor Etienne Barnard, for his guidance, suggestions and constructive criticisms, which have helped me enormously while working on this thesis.

I would also like to thank my sponsor, Namibia-Finland Forestry Program and my employer, The Public Service of Namibia through Ogongo Agricultural College.

I am also indebted to the entire Information and Communication Technology (ICOMTEK) division of the Centre for Scientific and Industrial Research (CSIR), in particular the e-government team which consisted of Hina Patel, Dr Louis Coetzee, Louis Joubert, Riette Easton, Mathabo Nakene, Soogandhree Naidoo, Natasha Govender and Sabeeha Hamza.

Besides my supervisor, the following people have help tune my English and my logic up: My wife, Mrs E-L Kafidi, Dr. E.N. Mvula, Dr. K T Kafidi, Dr. H B Winschiers, and Ms K Ndakunda. A big ‘thank you’ to all of them.

2. HUMAN COMPUTER INTERACTION

In the developed world, almost everybody has come into contact with computers in one way or the other. But, why do people use computers? Consider a typist, who uses a computer because she wants to produce some text onto a piece of paper. Her job is made easier, efficient and more enjoyable when using a computer (as compared to a conventional typewriter or writing manually). Thus, she prefers to use a computer. The idea behind Human Computer Interaction (HCI) is to make computers enjoyable, easy to use and useful. In this chapter, we will focus on conventional HCI - that is, HCI for users of computers such as desktop systems or personal digital assistants. In the next chapter, we will consider two telephone-based interfaces, where the interaction is mediated by the telephone channel.

During the early 1950s, computers were not so common because they were difficult to use, they were very expensive and were limited to a small set of tasks. Nowadays computers are cheap, easy to use, often small in size and can do a wide range of tasks such that they can be used by almost every literate person. Much of this progress can be attributed to developments in the field of Human-Computer Interaction.

There is no internationally agreed definition of HCI. But several definitions can be found in a number of sources. The most accommodating and convincing definition can provided by the Special Interested Group in Computer Human Interaction (SIGCHI) Web site [Hewlett 97], which states that: "*Human-Computer Interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use with the study of major phenomena surrounding them*." Put simply, HCI is the study of the interaction between people and computers. This study deals with the design, implementation as well as the evaluation of computing systems for human use. The main goal of HCI is to enhance aspects such as usability and safety without compromising the functionalities of the systems.

HCI has its roots in the "*user interface*" concept. The user interface (sometimes called Man-Machine Interface or MMI in short) is concerned with the aspects of a system that are directly accessed by the user of that system.

HCI is a multidisciplinary field, which makes use of ideas derived from diverse fields such as computer science, cognitive psychology, social and organizational psychology, ergonomics and, to a lesser extent, linguistics, artificial intelligence, sociology and anthropology [Preece 94, pp.34-37]. Computer Science provides information on capabilities and capacities of the existing technology as well as the techniques for the development of tools and methods for interface design. Cognitive psychology provides the relevant psychological findings and principles of human performance, which are then applied to the design of user interfaces. These findings include predictive models, guidelines and empirical methods. Social and organizational psychology helps in understanding structures and functions of organizations and how people interact among themselves within the organizations. This in turn provides the basis for development, introduction and the use of new technology in organizations. Ergonomics provides information on capacities and capabilities of users with emphasis on the effects of their physical aspects.

In studying HCI we look at, among other factors, human information processing, languages, communication and interaction, as well as physical human factors (ergonomics). The technology side encompasses the input/output facilities, dialogue technologies, dialogue genre or style, graphics and the dialogue architectures. The next sections discuss the human, the computer and the interaction in that order.

2.1 THE HUMAN

On the human side, there is a school of thought that models a human being as an information-processing machine. We will now look at the human side, guided by this model. In doing so, three human processes are of interest to us: how information is exchanged (input and output), how it is processed and lastly how it is stored and retrieved.

2.1.1 Information Exchange

Humans exchange information through seeing, hearing, touching, tasting and smelling. The last two do not play a major role in human-computer interaction. We will therefore concentrate on seeing, hearing and touching.

As soon as a human being sees (**vision**) an object, he can develop four major perceptions on that object, viz. the colour, the brightness, the size as well as the distance between him and that particular object. Colour is determined by the wavelength of the spectrum, the brightness of the colour and the amount of whiteness on that colour. Brightness on the other hand is the sensation that enables an observer to distinguish differences in light intensity per unit area.

By seeing two objects overlapping, a human will deduce that the object that is partially covered is at a further distance than the other one. Our knowledge of sizes of known objects also gives us a clue on how far the object is and vice versa. For example, seeing a diminished figure of an airplane enables a person to deduce that it is very high up. Conversely, knowing the distance between him and the object enables the person to judge how big the object is, just by looking at it.

A human differentiates sounds (hearing) by their respective frequencies and loudness by their amplitudes. The ears are also able to filter targeted sounds from background noise, hence our ability to concentrate on a conversation within a noisy group. Because our ears are a distance apart, we are able to determine the distance from

the source of sound by analysing the difference in the pair of sounds received by the two ears.

Unlike hearing and seeing (using two ears and two eyes respectively), the sources for touching stimuli (also called haptic perception) are not localized. There are millions of sensory receptors all over the human body, and different parts of the body have different sensitivities. There are *thermoreceptors* for heat and cold, *nocice receptors* for intense pressure, heat and pain as well as *mechanoreceptors* for pressure. Mechanoreceptors are the most relevant to the interaction we are dealing with. Mechanoreceptors in turn are divided into two categories, *rapidly adapting receptors* for immediate, short-term pressure and *slowly adapting receptors* for continuous, extended pressure.

The other issue related to touching is *kinesthesis*, -- the positions of the body parts with respect to the body itself. A human knows the position of his body parts because of the receptors at his joints. These receptors are divided into three categories: *rapidly, slowly* and *positional adapting*. Positional adapting is very important when it comes to touch-typing.

2.1.2 Information Processing

A human processes information either by reasoning or by problem solving [Dix 98, pp. 36-43]. Reasoning is a process where he uses knowledge to draw conclusions. It can either be done deductively, inductively or abductively. Problem solving is a process of using knowledge to find a way out of an unwanted situation. Three popular theories of problem solving are the *Gestalt, Problem Space* and *Analogical-Reasoning* theories. Gestalt theory stipulates that problem solving is both productive and reproductive. In other words, when solving a problem, we not only reproduce what is in our memory, we also produce something extra and store it in our memory. Problem Space theory stipulates that problem solving is but moving a problem from one state to another within the same space. All we need to know is the difference between the initial and the final state, and how to bring about that difference. According to the analogical-reasoning, we solve problems by identifying similar problems with known

solutions. We can then draw an analogue and apply that same solution with appropriate modifications.

2.1.3 Information Storage

The human being stores information in memory. This memory is classified into three categories, namely *Sensory*, *Short-Term* and *Long-Term* memory [Dix 98, pp.26-36].

Sensory memory is the most temporary form of memory, where information is stored for a very short period, up to 0.5 seconds. Sensory memories are further divided into categories, depending on which receptors provide the stored information: *iconic* for what we see, *echoic* for what we hear and *haptic* for what we get from touching objects. Playback of information from sensory memories is possible, provided that it is done before the content is discarded. From sensory memory, information is either discarded and lost forever, or passed to the short-term memory (when we pay attention to something) for further processing.

Short-term memory is the human version of Random Access or scratch pad memory. Though it has limited capacity, it has a very high accessibility rate, with access times of about 70 milliseconds. Capacity is measured by how many items from a given sequence or from a random set one can remember, the more items one can recall, the bigger is the capacity. From here, information is passed into long-term memory for permanent storage. The so-called *recency effect* states that it is easier to remember the items encountered most recently.

Finally, long-term memory is our main storage and it is where what we call ‘knowledge’ is stored. Its capacity is huge (for practical purposes, unlimited) but its accessibility is slow, with typical access times on the order of seconds. Decay or loss of information from long-term memory is slow, and the *recency effect* is less pronounced in long-term memory. In fact information stored last week and last year are often accessed with the same ease. Long-term memory is subdivided into *episodic* memory, which keeps a structured record of events in serial order and *semantic* memory which stores facts, concepts and skills. If we are to recall a past

event, we retrieve it from episodic memory, while if we are to deduce something from past experience, then we have to put semantic memory to work. There is a continuing debate as to what exactly happens when one fails to access a piece of information: does it mean: "*the information is not there"* or "*the information is there but just too difficult to access"?*

Storage of information into long-term memory is done from short-term memory in a process called rehearsal. How much we can store is affected by how the information is structured at presentation, how familiar the person is with the context of the information and how concrete the information is.
Retrieval of information from long-term memory is done either by reproducing a chunk (recall) or by deducing some knowledge from information that is stored (recognition).

Information is lost by either long-term fading (decay) or by interference (replacement). These processes are highly affected by emotional factors.

2.1.4 Applicability to the field of HCI

We have briefly reviewed the capabilities and limitations of humans which will affect their interactions with the computer. With this in mind, we can design interactions where these capabilities are enhanced, limitations are minimized and, where possible, compensation is provided. Here are some examples of straightforward guidelines that can be deduced from the principles of human perception and cognition:

- Since 8% of males and 1% of females are colour blind, certain colours should not be overused as a means of contrasts.

- Humans can gather information from sound without necessarily concentrating on the source. Therefore we can use sound to:
    - draw attention (*read this popped-up message*),
    - convey system status information (*a process has finished*),
    - confirm (*yes, your input has been accepted*)

- convey navigational information (*you have moved to another window*),

without overloading the human information input capacity.

- Taking into consideration the *recency effect*, it is important to make sure that we give information exactly when it is most needed. It is not advisable to provide information and make person wait before he can make use of it.

In other cases, it requires significant expertise to apply cognitive principles to HCI. Fortunately, guidelines to guide in the development of interactions have been developed, models to build on have been created and empirical techniques for evaluating the interaction have been proposed. This means designers can incorporate information about human cognition without becoming experts in the field of cognitive psychology themselves.

2.2 THE COMPUTER SYSTEM

After looking at the human as a partner in the interaction, it is now the computer's turn to be examined with respect to HCI. Although computer systems vary according to the tasks they are meant to fulfil, we will consider a typical computer system that has means of communicating information to and from the user.

Information processing by a computer system can be performed as either batch or real-time processing. For batch processing, the user feeds in all the necessary information required to complete a task in a single interaction. The system is then left to complete the task without additional input from the user. This processing style requires relatively little interaction and does not form part of the subsequent discussion. Real-time processing occurs when the user and the system exchange messages/information continuously while fulfilling a task. The main interest to us is how (rather than what) the information is exchanged between the system and the user, and how the different interfaces make it easy for that process to take place. We will first look at how the computer system exchanges information by looking at both input and output devices. We will then proceed with information processing and lastly how information is stored by a computer system.

2.2.1 Text entry devices

2.2.1.1 Keyboard

The keyboard is most commonly used for text and command entry. There are three popular keyboards in common usage, namely the QWERTY, DVORAK, and chord keyboards. We will briefly look at these designs and thereafter, look at others that are emerging on the market

QWERTY

The QWERTY keyboard is based on the (outdated) mechanical typewriter. The original arrangement of the keys was done in such a way that common two-letter combinations are placed on opposite sides of the keyboard. The typewriter was mechanical and required the mechanical motion of keys. It was therefore important that common combinations of keys do not interfere with each other. As typewriters were replaced by computers, many typists had already learned to use the QWERTY keyboard and the designers used that fact to the advantage of both the system vendors and users. Hence, the QWERTY layout survived beyond the context of its original design.

DVORAK

The DVORAK keyboard (named after its inventor, Dr A Dvorak) was designed as an alternative to the traditional QWERTY keyboard, designed in order to maximize typing speed. Its layout is arranged such that the most common letters are placed as home keys. Though the claim that it has increased the tying speed is still debatable, it is generally agreed that Repetitive Strain Injury, to which QWERTY keyboard typists are subjected, has been reduced [Bigler 03], [Liebowitz]. But because of the cost involved in retraining users, the DVORAK keyboard did not reach a wider market as expected.

Chord

Chord keyboards are smaller in size and have very few keys in comparison to the other two. Instead of using a separate key for every letter, it uses simultaneous key presses, in much the same way as the ALT and SHIFT keys are used in the conventional keyboards. With this keyboard, the fingers' travelling distance is

drastically reduced, since many combinations can be made out of few keys (e.g. with only 5 keys, 31 different characters or commands can be typed). The Chord keyboard is also very portable and requires limited space to operate. However, it requires intensive training before usage and even after training, typing speed is slower than with the other two designs.

Besides these keyboard layouts, a number of designs have been introduced commercially but failed to flourish, and some are still emerging into the computer market. Top of the list is probably the Adjustable-Spilt keyboard, which is designed so that they can be split horizontally or/and vertically while still maintaining the same layout (QWERTY or DVORAK). There are also some designs that cater specifically for disabled people, again with either DVORAK or QWERTY layouts.

2.2.1.2.Other text entry devices

Handwriting recognition

Handwriting recognition enables the computer to recognize characters and other symbols produced by a human hand. It generally uses a digitization tablet, where the continuous strokes of the pen are converted into a series of coordinates [Drissman 97]. A screen is incorporated on the tablets, so that the information captured can be displayed in the form of an electronic document. The information captured can also be stored in the same form for later use, but is more useful when it is converted to typed text. This technology has the advantage that it does not need special training, as handwriting is a common skill. But variations in handwriting styles and the “effect of the previous letter” on the current one (similar to the co-articulation effect in speech) are obstacles to accurate recognition. The speed at which users write is also relatively slow in comparison to typing. Segmentation of handwritten letters into single letters is very difficult, mostly due to the inconsistency in the human handwriting. There have been some attempts to recognize pairs of letters, whole words and even phrases, but none of these approaches has produced significant improvements in the accuracy of the handwriting recognition.

Gesture Recognition

Gesture Recognition allows a human to interact with a computer using hands and/or facial movements. It is commonly used when other modes, typically typing, cannot be used efficiently. Vision systems or data gloves are used to capture user input and these are still very expensive. As with handwriting recognition, systems also experience difficulties with segmenting the gestures.

Speech Recognition

Speech recognition enables a computer to understand and respond to a human voice. A typical system matches speech input against a stored representation. Most systems use spectral representations, and models such as templates or hidden Markov models to represent spoken elements. These systems are classified as speaker dependent or independent, and as discrete or continuous speech input systems. Other important characteristics include the recognition accuracy rate and the size of the vocabulary of the recognizable words (small, medium or large). This technology is intimately related to one of the two interfaces (together with touch tone) to be compared later, and will be dealt with in more details in the next chapter.

2.2.2 Positioning and pointing devices

2.2.2.1 The Mouse

Although pointing devices such as light pens, joysticks and graphics tablets have all been used extensively, they are currently totally dominated by the mouse, which was introduced into the mainstream computer market with the Apple Macintosh in 1984 [Brain 03]. Its success was mainly due to its inexpensiveness when compared to the earlier pointing devices, its size which needs little desk space and its utility in conjunction with GUI interfaces, in particular, window-based interfaces. The wheeled mouse has a ball in its belly, which remains in contact with the table surface and rotates when the mouse moves round. There are two rollers inside the mouse that are in constant contact with the ball. These rollers detect and transfer movements in the y- and x-directions to the computer respectively.

The popular but venerable wheeled mouse is in danger of being totally replaced by the optical mouse, developed by Agilent Technologies and in use since 1999 [Brain 03]. It works by gathering a sequence of images, with the aid of the light-emitting diode, at extremely short intervals. The images are sent to a Digital Signal Processor, which in turn determines the position of the mouse by processing the variation of the successive images it receives.

2.2.2.2 Other Devices

The mouse is the most common pointing device, but some alternatives have been designed. Below are some other common pointing devices:

A *Trackball* is best described as an upside-down mouse [Gilman]. The rolling ball on top of it is housed in a static entity. The movements of the ball are detected and transferred as in the case of the mechanical mouse. But, unlike the mouse, it requires less space. Different modes of manipulation (pointing and clicking) are performed separately. The size of the ball varies from a marble to a cue ball, the bigger the ball the more effort is required to rotate it. Buttons for single click, double click and right click are usually placed on the side of the ball.

A *joystick* consists of a stick or a grip sticking out of a small plastic base. Moving the stick causes corresponding movements of the cursor on the screen. There are two types of joysticks, *absolute* and *isometric*. The absolute joystick emphasizes movements: the position of the stick and the movements thereof corresponds to the positions (and movements) of the cursor on the screen. The isometric joystick is keyed to the pressure applied to the stick, which it converts to the cursor's velocity. The more pressure applied, the faster is the cursor's movements. The buttons are usually on the side, in front (like a trigger) or behind the stick.

A *Touch Screen* contains tiny sensors to detect pressure from pointing devices or fingertips. Manipulation is done by pressing the sensors which react like a traditional mouse when clicked, double clicked or dragged. As with other pointing devices, accompanying software allows the user to customize the settings. Touch screens are common on laptops.

A *Head Tracking Mouse* works by transmitting a signal from a monitor and tracking a reflector placed on the user's head. It allows the person to control the movements of the cursor using only the movement of her head. When the head of the user moves, it translates to the direction the cursor will move. Unfortunately, the head-tracking mouse does not offer clicking functions

Eyegaze systems allow the user to interact with the computer using her or his eyes. Cameras that focus on the user's eyes are mounted on the monitor. The cameras determine where the user's eyes are focused and the 'gaze point cursor will be placed on that spot. Mouse clicks are obtained by different behavior of the eyes (eye dwelling, slow eye blinking) or through some other hardware. Eyegaze systems are helpful for disabled people who cannot make use of their hands during the interaction.

2.2.2.3 Keyboard-based positioning devices

Besides the traditional arrow keys and the switching ON/OFF of the numerical keypad to transform it into cursor controllers, some software (e.g. Mouse Keys) transforms the numerical keypad into a directional mouse. When installed and activated, some keys work like directional keys while others assume the role of different clicking functions.

2.2.3 Soft Copy Output Devices

2.2.3.1 Cathode Ray Tube (CRT)

A CRT consists of a cathode, a pair (or more) of anodes, a phosphor-coated screen, a conductive coating and some coils [Brain 03]. When the cathode element is heated inside a vacuum tube, it emits a ray of electrons. The electrons are focused and accelerated towards a phosphor coated flat screen by two different anodes. The electrons are steered by steering coils. On being struck by the electrons, the screen glows.

To display a black-and-white image, the screen is coated with a white phosphor. The beam paints the image along the horizontal lines and proceeds downwards. For colour display, there are three different beams to generate red, green and blue

components of image. The screen is also coated with three different stripes for the three colours. The vacuum tube contains a shadow must, with very small holes aligned with phosphor stripes on the screen. To display a blue colour, the blue beam is fired onto the screen (the same applies to red and green). To display white, all the three beams are fired, and for black, all the beams are switched off.

2.2.3.2 Liquid Crystal Display (LCD)

LCDs work on four principles [Tyson 03], viz. the ability of some transparent substances to conduct electricity, the ability of electric current to change liquid crystal structures, the ability of the liquid crystal to transmit and change light and the polarizable nature of light. An LCD is a sandwich of glass panes, a liquid crystal and some transparent electrodes. Popular liquid crystal materials include *super twisted nematics*, *dual scan twisted nematics, ferroelectric liquid crystals* and *surface stabilized ferroelectric liquid crystals*. Although most LCDs are reflective and do not require their own light source, many computer monitors make use of fluorescent tubes on top, alongside and sometime even behind them. In such cases, a diffusion panel is used to redirect and scatter the light rays evenly. LCDs are categorized as either passive or active matrix.

In a passive matrix, pixels are charged by a simple grid, which is constructed from two substrates (for rows and columns, respectively) made of mostly indium-tin oxide. The charges to the pixels are supplied and controlled by integrated circuits connected to the substrates. A polarizing film is attached to the outer side of each of the two substrates and the two sandwiches containing the liquid crystal material. For the pixel to be turned on, a charge is sent by the integrated circuit to the correct column of one substrate and a ground activated on the correct row of the other. At the point where this row and column intersect, a voltage will be delivered that will align the liquid crystal. This structure's main disadvantage is its slow refreshing rate.

An active matrix relies on thin-film transistors arranged in a matrix on a glass substrate. To turn on a particular pixel, the relevant row is turned on and a charge is sent to the relevant column. The transistor in that row and column will receive the

charge and will retain it until the next refresh cycle. With the aid of the liquid crystal, the amount of light is regulated and hence a variable grey scale is created.

For colour display, there are sub-pixels with red, green and blue colour filters for every spot, rather than a single pixel. The combination of these colours can be used to create 17 million different colours with current technology.

2.2.3.3 Speech Synthesis (Text-To-Speech)

Speech synthesis (or text-to-speech, as it is popularly known) is the process of converting text to spoken language. The process involves breaking down words into smaller components called phonemes, analysis of special text such as punctuations, numbers or currency and then the generation of digital audio. Although it is not difficult to understand the voice produced by modern speech-synthesis systems, they do not sound particularly natural. Speech synthesis systems are a popular alternative where other forms of communications are not possible, for example for blind users. Speech synthesis will also be discussed in the next chapter as part of a telephone-based interface.

2.2.4 Hard Copy Output Devices (Printers)

2.2.4.1 Dot matrix printers

Dot matrix printers are mechanical. To produce an image on a paper, the paper is passed between a carbon-coated ribbon and an anvil. A series of pins, each representing a character is then used to strike/press onto the ribbon, thereby producing an image on the sandwiched paper.

2.2.4.2 Inkjet/DeskJet printers

An inkjet printer produces an image by placing very tiny droplets (dots) of ink onto the paper. The quality of the image is enhanced by the facts that the dots are very small (with a typical diameter of 60 microns), and are accurately positioned (up to 1440 x 720 dots per inch). Dots can be of different colours for colour images. The ink is sprayed onto the paper from a print head that contains the spraying nozzles. The ink is contained in printer cartridges.

2.2.4.3 Laser printers

Laser printers use static electricity: an electrostatic image is first formed on a photoconductive drum assembly, and this image attracts carbon particles from the toner. The paper, previously charged, is then rolled under the drum at the same speed as the drum. The paper will attract the carbon particles from the drum assembly thereby duplicating the image. The paper is then passed through the fuser that melts the carbon particles and fuses them with the paper.

2.2.5 Information Storage

Information on a computer is stored in memory. The computer memory can be divided into two categories, temporary (volatile) and permanent (non-volatile). Temporary memory stores the information as long as the computer power is on. When the computer is powered off the information is lost. The most common temporary memory is Random Access Memory (RAM). Permanent or non-volatile memory keeps the information stored even after the power has been switched off. Permanent semiconductor memory is known as Read Only Memory (ROM); there are also various forms of disk- or tape-structured permanent memory, as discussed below.

2.2.5.1 Structure and Characteristics of Temporary Memory (RAM)

RAM is the location on the computer where the currently used data is stored. They get their name from the way they are accessed, random, as opposed to Sequential Access Memory (SAM). Random access implies that it is possible to access any memory cell if the address of that cell is known - unlike SAM where you have to 'page' through until you are pointing at the right place where the desired data is stored.

The most common form of RAM today, Dynamic RAM or DRAM, is instantiated by integrated circuits composed of transistors and capacitors [Tyson 03]. A memory cell, which represents one bit of data, is formed at each intersection of a row and a column by one transistor and one capacitor. Memory cells are addressed through their

columns and rows. The capacitor holds the charge of that particular cell. The transistor acts as a switch through which the circuitry can access the capacitor. After charging, the charge on the capacitors decreases as the capacitors do leak continuously. Therefore, they have to be recharged at regular intervals - hence the name “dynamic” RAM. When in use, a charge is sent through the relevant column to activate the transistors. For writing, the appropriate row will contain the state of the capacitor and eventually pass this information onto the capacitor, charged or not charged. For reading, the sense-amplifier determines the level of charge on the capacitor. If the level is more than 50% of the original charge, the bit is deemed to contain a 1, else it contains a 0.

Static RAM or SRAM is slightly different from DRAM. SRAM is faster and more reliable than DRAM. A memory cell takes 4-6 transistors and they do not have leaky capacitors. The transistors themselves are switches always in flip/flop position depending on the value they are representing. They are termed static because they do not need to be refreshed continuously. SRAM is mainly used for cache because of its greater complexity (hence price) and higher typical access rates.

Besides DRAM and SRAM, other types of RAM on the market include FPM DRAM, EDO DRAM, SDRAM, DDR SDRAM, RDRAM, Credit Card Memory, PCMCIA Memory Card, CMOS RAM and VRAM.

Fast Page Mode Dynamic Random Access Memory was the original form of DRAM. When accessing a bit of data, the system has to wait for the process of locating that bit’s column and row before starting to locate the next bit. Extended Data-Out Dynamic Random Access Memory (EDO DRAM) continues with the next bit without waiting for all of the processing of the first bit. Soon after locating the address of the first bit, EDO DRAM begins looking for the next bit. Synchronous Dynamic Random Access Memory (SDRAM) takes advantage of the burst mode concept to greatly improve performance. It stays on the row containing the requested bit and moves rapidly through the columns, reading each bit as it goes. Double Data Rate Synchronous Dynamic Random Access Memory (DDR SDRAM) is just like SDRAM except that is has higher bandwidth, meaning greater speed.

Rambus Dynamic Random Access Memory (RDRAM) is a radical departure from the previous DRAM architecture. The major difference between RDRAM and the traditional DRAM is its use of a special high-speed data bus called the Rambus channel.

Credit Card Memory is a vendor-dependent self-contained DRAM memory module that plugs into a special slot for use in notebook computers. PCMCIA Memory Card is also a self-contained DRAM module and is mainly used for notebooks. But unlike Credit Card Memory, they are not vendor-dependent and should work with any notebook computer whose system bus matches the memory card's configuration. CMOS RAM is a term for the small amount of memory used by the computer and some other devices to remember things like hard disk settings. Video RAM (VRAM) also known as Multi Port Dynamic Random Access Memory (MPDRAM), is a type of RAM used specifically for video adapters or 3-D accelerators.

Memory capacity has increased considerably during the past three decades. Nowadays the standard memory capacity is at least 128MB. Accessibility to RAM is relatively fast. Data can be accessed at 200ns and can be transferred up to a rate of 10MB per second.

2.2.5.2 Structure and Characteristics of Permanent Memory (ROM)

Solid-state ROM also consists of integrated circuits but instead of using transistors and capacitors, the circuitry connection is done by diodes [Tyson 03]. The charge is sent through the column and for the 1-value bits, the diode is present and hence electricity is conducted. For a 0-value bit, there is no diode to establish the connection between the row and the column and therefore no charge is flowing. The principal characteristic of standard ROM is that it cannot be reprogrammed or rewritten. In other words, no alteration is possible once the data has been written to the chip.

The creation of a single ROM is quite expensive and time-consuming. This is why developers invented a Programmable Read Only Memory (PROM), which is normally

bought blank and can be coded using a tool called a “programmer”. Initially, each row-column intersection is connected by a fuse (instead of a diode), thus setting the value to 1. When writing the data onto the PROM, some value-1 bits are changed to 0. To do this, a specific amount of charge is sent to a particular memory cell. The charge will burn the fuse, and break the connection at that intersection. Although PROMs are more flexible than standard ROM, they can only be altered once because the fuses are irreplaceable.

Additional flexibility is offered by Erasable Programmable ROM (EPROM). EPROM can be rewritten several times. Erasing is done by a special tool that emits certain frequencies of Ultraviolet light. EPROM also consists of a grid of rows and columns but the connection between each row and any column is done via two transistors, called the control and floating gates, respectively. A thin oxide layer separates the two transistors. The floating gate is linked to the row via the control gate. As long as this link exists, the bit value is 1. To change the value to 0, a process called Fowler-Nordheim tunnelling has to take place. This process blocks the flow of electrons between the two transistors. To erase the EPROM, the connections between the floating and control gate pairs are re-established, thus setting all the values to 1. The alteration of EPROM also requires special equipment, and there is no way to erase just a portion or the chip. Electrically Erasable Programmable ROM (EEPROM) and Flash memory allow the user to erase a portion of the memory while keeping the rest intact.

Permanent memory is typically embodied in magnetic disks and optical disks. Magnetic disks include floppy diskettes and hard disks. Floppies are removable and easy to carry around but their capacity is very limited. A typical floppy’s capacity is 1.44MB. Hard disks are usually mounted inside the system unit and therefore non-removable, but have much larger capacity. A capacity of 40GB is typical. Accessibility and data transfer rates are slow compared to RAM. Data can be accessed within 10ms and is generally transferred at 100KB per second, which is more than a hundred times slower than with RAM.

Optical disks come in the form of Compact Disks, which include CD ROMs, which cannot be rewritten, and CD RWs, which can.

2.2.6 Information Processing

In a standard (Von Neumann) computer, the information is processed by a Central Processing Unit (CPU), typically embodied in a microprocessor. A microprocessor is a general-purpose computational engine, fabricated as a single integrated circuit. It consists of a Control Unit and an Arithmetic and Logic Unit. It performs four steps in executing an instruction [Brain 02]:

(i) *Gets the instruction from memory*

(ii) *Decides what the instruction means and arranges for the availability of data to be used*

(iii) *Performs the actual operation on the data.*

(iv) *Stores the result of the operation in memory or register.*

Microprocessors are characterized by their instruction set, their bandwidth and their clock speed.

Intel has been a leader in the manufacturing of processors, with AMD and to a lesser extent IBM contesting for a significant share on the market. The number of transistors on a single chip as well as the clock speed has increased dramatically. In 1972, Intel's 8008 processor had 3 500 transistors and a clock speed of 0,2MHz. 30 years later (2002), Intel's Pentium 4 had 55 million transistors and a clock speed of 3 000 MHz [Strandberg 03].

In HCI, we must understand how the limitations of such a computer can impact on its interactive usage. Such limitations usually fall into one of the four categories: Processing can be *computation bound* when the processor itself is the bottleneck, or *storage-channel bound* when either the storage media or transfer to and from the media determines the processing times. A process is *graphics bound* when the display of information is the slowest part of the computational pipeline, and *network bound* if information is transferred over the network, which restricts the overall processing rates.

Fortunately, processing power and storage capacity have increased considerably with a corresponding decrease in their respective costs. However, screen size, resolution and colour range have all increased, and sound and multimedia interaction have become common. Users continually expect more functionality from software, and larger storage capacities have made it possible to provide increasingly complex solutions. Thus, processing issues remain important in HCI, albeit less critical than in previous eras.

2.3 THE INTERACTION

For the purpose of this discussion, a *goal* means a desired output from a performed task. A *domain* means an area of expertise and knowledge in a real world activity. A *task* means an activity to manipulate concepts of a domain. An *intention* means a specific action required to meet a goal. An *interaction* is deemed to have taken place when a user carries out a task within a certain context by making use of a computer system. An *interface* is the 'medium' through which the users and the computer system communicate.

2.3.1 Interaction models

By modelling interactions, we can focus on a number of underlying issues in interaction systems without dealing with the full complexity of the role players therein. We will briefly discuss two models, Norman's execution-evaluation cycle, and the Interaction framework.

Norman's execution-evaluation cycle divides an interaction into two phases, the execution and evaluation phases [Dix 98, pp. 104-109]. These are jointly carried out in seven distinct stages as follows: The user establishes the goal; she forms an intention; specifies the sequence of actions; executes the actions in that sequence; perceives the system state after the execution; interprets the state of the system and lastly evaluates the systems with respect to the pre-defined goal and intentions. In describing the ease with which the interaction takes place, Norman came up with two terms, the gulf of execution and the gulf of evaluation. The former describes the

difference between the user's formulation of an action and the actions that can be performed by the system. The interface is aimed at diminishing this difference. Therefore, the less the difference, the more effective is the interface. The gulf of evaluation is, in effect, the difference between the expected state of the system and the physical presentation of the state. The interface should also present the physical state of the system so that it will be easy for the user to evaluate whether the goals are met. The major shortcoming with this model is that it only considers the user's perspective of the system.

The interaction framework divides the interaction into four components, the *user*, the *input*, the *system* and the *output*. The respective languages for these components are labelled *task, input, core* and *output,* respectively. Input and output components jointly constitute the interface. According to this model, the interaction takes place in a four-step cycle as follows:

- The user's goals are interpreted into the input language as stimuli for the input component.
- The input stimuli are then translated by the input component into the core language for the system stimuli.
- The system, after receiving the stimuli, will transform itself according to the instructions contained in the stimuli. The end of this transformation also marks the end of the execution cycle and hence the beginning of the evaluation cycle.
- The system translates its new state into the output language for the output stimuli. It is of utmost importance that the stimuli convey the state attributes as accurately as possible.

On receiving these stimuli, the output component translates them into the task language for the user stimuli. The user uses these stimuli to evaluate whether the system has been transformed into the desired state, and that marks the end of both the evaluation phase and the interaction cycle. The input-system-output communication is of less importance to the user, but not to the designer.

2.3.2 Ergonomics/Human Factors

Ergonomics deal with the physical characteristics of the interaction. In particular it deals with the design of controls, the physical environment of the interaction and the physical qualities and layout of the screen. It mainly focuses on the user's performance and how it is affected by the interface.

Controls can be arranged *functionally* where controls that perform related functions are grouped together, *sequentially* where grouping reflects the order in which they are accessed when completing common tasks or by *frequency,* in which case the most frequently accessed controls are also the most accessible.

The physical environment consists of the components such as the identity of the user, her physical location when using the system and issues such as visibility and accessibility of controls. Several generic guidelines apply. For example, all system controls should be accessible by an appropriate range of users, disabled users possibly included. Any user must be able to see critical displays. If the user sits while using the system, then back support should be provided, and if she stands, there must be room to move around. The period of standing should not be excessive. If the system is to be used for longer period, then some form of resting should be provided.

There are also some health issues to be considered. The temperature in which the system is to be used should be favourable to the user, too low or too high temperatures are detrimental to the users' health. Adequate lighting must be provided and the positioning of the source of light must be placed so as to avoid eyestrain and excessive reflection. While noise can be utilized as stimuli, excessive noise must be avoided. Constant usage of some equipment is also unhealthy to users, e.g. CRTs to pregnant women, and care must be taken accordingly.

Any colours used must generally be as distinct as possible. If colour is used as an indicator, other means must augment it to cater for colour-blind users. General colour conventions must be respected, e.g. red for stop or danger, green to go etc.

2.3.3 Interaction styles

2.3.3.1 Command Line Interface (CLI)

The command line interface is a means of expressing instructions using Function keys, single characters or words. It offers direct access by avoiding hierarchies and offers options for parameters e.g. in cases of repetitive instructions. The problem with CLI is that, it is difficult to learn and the commands are usually not consistent.

When designing commands for the CLI, they must make sense in the user's (and not just the designer's) language.

2. 3.3.2 Menus

With menus, options are displayed on the screen and selections are made by means of a mouse, touch screen, key on the keyboard, or similar device. Menus are easy to use, as they require minimal memorization. On the other hand, menus are in most cases hierarchical and all options cannot be displayed at the same time. When naming options, it is important that the names are meaningful to the user.

2. 3.3.3 Natural Language

Natural language is arguably the most attractive and useable interface. Unfortunately, natural language tends to be ambiguous to the computer. To cater for the ambiguity, the domain is usually restricted to given terms and/or phrases. To the user, natural language supports creativity in expression but the computer on the other side of the interaction must receive clear and unambiguous instruction. Hence this form of interaction places significant demands on the designer.

2. 3.3.4 Question/Answers and query dialogue

A dialog provides input to a specific domain by asking questions to which the user provides yes/no, multiple choices or other forms of guided answers. It is easy to learn, but very limited in its functionalities. Query language is used to construct queries on a database. The instructions are very similar to natural-language phrases. For effective usage, the user must possess some appropriate experience.

2. 3.3.5 Forms and spreadsheets

Forms are displayed and fields where the data is to be filled in are left blank for the user. Forms are also easy to use and very flexible in navigation. They are primarily used for data collection and offer a means to make corrections.

Spreadsheets are a variation of forms. They are composed of cells, each able to contain different information. Data can be entered and altered in any order.

2. 3.3.6 Point and Click

A point and click interface allows the user to point at certain item and click on it, to perform a chosen action. It is mainly used in mouse-based systems and systems with touch screens, and is also popular in hypertext where the user can click on highlighted words, maps or iconic buttons.

2. 3.3.7 Three-dimensional (3D) interfaces

The simplest form of three-dimensional interface is where items (windows, buttons) are given 3D appearance using shading. But a more complex technique makes use of size, lighting and occlusion to provide a sense of distance. Objects are shrunk, raised or flattened accordingly.

2. 3.3.8 WIMP

WIMP stands for windows, icons, menus and pointers and its parts are called widgets.

A *window* is an area on the screen that behaves like an independent entity. It has a title bar that identifies it and can host text or graphics and can be moved or resized. Multiple windows can be displayed and multiple threads are therefore supported. Windows can be tiled or cascaded.

Icons are small pictures that represent windows, files, folders, drives or network connections. Shrinking a window into an icon is called iconification. Icons can be highly styled and may be represented by various symbols.

Pointers are a very important component of the WIMP interface as they provide the means of input from the user. A mouse is mainly used but alternatives such as a joystick or a trackball are also common. The pointing device is represented by a cursor on the screen. Cursors in turn can have different shapes but they all have a hotspot, i.e. the effective portion. The hotspot must be clearly identifiable to the user.

Menus provide the list of possible actions that can be performed on the system. A pointing device is normally used to navigate through menus and to make choices. Selection requires an additional action e.g. a click or a stroke of a key on the keyboard. Where there are many options, pull-down or popup menus must be used to conserve space. The principal problem with the design of a menu is 'what items to include at what depth'.

Buttons are individual and isolated mini-areas within a display. Pressing a button invokes a specific action. Buttons can be used to toggle between states. When deciding on where to place buttons on the screen, buttons (like other controls) can be grouped functionally, sequentially or according to the frequency of use.

Toolbars are collections of small buttons for commonly related used functions. Sometimes, the designer chooses the composition of the collection, but the user can also be allowed to assemble her own collection when possible.

Palettes are sets of possible active modes on the system. Dialog boxes are information windows to convey messages and to draw users' attentions. Sub-dialogs can also be invoked within a main dialog for a specific task, and closed as soon as that particular task is completed.

2.4. INTERACTIVITY

Interactivity is the defining feature of most user interfaces. It determines the feel of the WIMP, since WIMPs tend to be composed of the same components but behave differently in different environments. In old computer systems, the order of interaction was determined by the computer system. Nowadays, except in a few cases (like

modal dialog boxes), the user must determine the order of the interaction. A good interface design should maintain flexibility on the order of the interaction.

Interaction is strongly affected by social and organizational issues. Though mostly beyond his control, the designer must be aware of these issues and where possible, take appropriate measures. The introduction of new technology must involve users too, not only managers and designers. A user may need to acquire additional skills in order to use the new technology; she should be appropriately motivated and her privacy must be preserved.

CHAPTER 3. SPEECH RECOGNITION SYSTEMS

3.1 INTRODUCTION

The facilities through which information is accessible in the developed world are continually changing, e.g. from desktop computers and telephones to mobile computing devices such as Personal Digital Assistants (PDAs), tablet PCs, and next generation phones. But in the developing world, desktop computers and standard telephones are still the main forms of computing and information access. Telephones in particular are widely available and their distribution keeps spreading. It is therefore imperative that user interfaces should be developed that makes information accessible via telephones in developing countries.

In order to develop and successfully implement telephony interfaces, a thorough knowledge of the underlying technology is required. The designer should know the capability and limitations on the existing technology. Since interfaces based on speech recognition are currently the state of the art, the aim of this chapter is to discuss the characteristics and functionalities of the existing speech recognition technology. A brief description of the principles of Dual Tone Multi Frequency (DTMF) systems - the previously dominant paradigm for telephone interfaces - is also presented towards the end of the chapter.

3.1.1 Speech Technology

Speech technology consists of several components, including speech synthesis (or text-to-speech) and speech recognition systems. The aim of these technologies is to enable the users to communicate with a computer system using speech as output (in speech synthesis) and input (in speech recognition). The task of the technology is to act as the ‘translator’ between the computer system and the user, to make sure that the two parties understand each other’s messages. Below is a brief description of the basic concepts on speech synthesis. Speech recognition will be covered in detail in the next sections.

3.1.2 Speech Synthesis Systems

Speech synthesis systems produce audible speech from computer-readable text. This is done either *by concatenation* or *by synthesis-by-rule*. Concatenation records human speech as phrases or words, and saves these utterances. The phrases or words to be spoken are then assembled into sentences in the right order while filling in all the missing information (numbers, names, etc). As an example, to ask for a confirmation of a certain telephone number, the question may be given as: “Did you say 123456?” In this case, the phrase “Did you say” as a whole may have been pre-stored. When in use, the system will play that phrase and only fill-in the number 123456. The smoothness of the sound is usually poor and the output sounds artificial.

Synthesis-by-rule does not pre-record any human voice. Instead, words and sentences are constructed using the rules of acoustic phonetics and the context of sentences. This approach has more potential flexibility than concatenation, but the output of current systems is very artificial when compared to the human voice. Currently, the best synthesis quality is achieved with concatenative systems, and synthesis-by-rule is used mostly in applications where the storage of large databases is not feasible.

Speech synthesis plays an important role in the implementation of telephony interfaces. They determine the sound quality of the prompts which the user hears. But in the absence of indigenous speech synthesis systems, one is forces to pre-record the prompts if prompts in the indigenous languages are to be produced. But it is worth noting that if the same techniques are used in preparing the prompts, it will result in prompts of the same quality. Thus, their effect is unlikely to favour any of the two interfaces.

3.2 SPEECH RECOGNITION (SR) SYSTEMS

Simply put, speech recognition is a process aimed at converting speech from its acoustic form into its text equivalent. Over the years, the keyboard, the mouse and remote controls have dominated human interactions with the computer. As a result, interacting with computer systems has been mostly limited to users who have had good technological exposure. Other users’ interactions with the computer would be

rather too slow as they are not familiar with the keyboard/mouse/remote control layouts. Speech recognition has the potential to address this problem, because speech is a widespread human capability.

Speech recognition is used for a variety of applications, including command-and-control, data entry, data access and dictation [Markowitz 96, pp181-182].

The research and development of speech recognition systems has received significant attention for about 40 years. Although much progress has been made, 100% accurate recognition and understanding (by the system) of naturally spoken speech remains far beyond the reach of technology. The Department of Defence of the United States of America has been the pioneer in funding much speech recognition research through the Defence Advanced Research Project Agency (DARPA). A number of research institutions in the United States took the lead in this research, particularly the Carnegie Melon University (CMU) and the Massachusetts Institute of Technology (MIT). In the early 1970, systems were developed which could recognize speech from specific users speaking in a particular manner. The most successful one at the time was CMU's HARPY. HARPY could recognize complete sentences subject to some grammar constraints. During that period, speech recognition systems were expensive and they required significant processing power, which was not affordable then. By the 1980s, the accuracy of speech-recognition systems had improved drastically, while processing power increased with a decrease in cost.

As an input interface, speech offers fast input (most people talk faster than they type or write), a hands-free environment, mobility, over-the-telephone interaction as well as intuitive and natural communication. While the four main types of speech recognition applications (command-and-control, data access, data entry and dictation) differ in their structures and uses, they use similar technologies and approaches.

Some speech recognition systems require that the user should 'train' them before use. That means the only user who can use these systems is the one who is 'known' to the system. These systems are called speaker dependent. Other systems do not

need such training, and are consequently immediately usable by any user - the so-called speaker independent systems. Furthermore, the input speech can be categorized as continuous or discrete speech, depending on whether or not the user is forced to make forced pauses between words.

3.2.1 The Speech Recognition Process

The speech recognition process can be divided into two major sub-processes viz. pre-processing and recognition. Speech input is produced in an analogue waveform. Before it can be used by a computer system, it has to be digitised. Pre-processing is the component that converts the speech input from analogue to digital form, and computes a useable representation of the digital data. Recognition is the process that takes the pre-processed input and compares it with some stored models in order to determine the meaning of the input.

3.2.1.1 Pre-processing

Pre-processing encompasses all the events that happen immediately after the speech input enters the microphone until just before the actual comparison takes place. These include capturing of the input, digitisation, spectral representation and segmentation. As soon as the speech input has been captured, it has to be converted from analogue to digital form. For the conversion to attain the desired speed and accuracy, it has to include all critical data, remove redundancy, decrease the noise and distortion and make sure that it does not introduce some other forms of distortion to the same data. Most systems ‘sample’ the input further into frames. The processor will then capture the acoustic patterns for each frame and record the changes in successive frames. This process is called spectral analysis. There are mainly two approaches to spectral analysis:

The bank-of-filters approach employs a set of filters to dismantle the sample into frequency bands. The frequency bands are then converted into arrays of acoustic parameters.

Linear Predictive Coding (and similar methods) estimate model-based acoustic parameters of the incoming frames, sometimes incorporating the parameter values of

the preceding one. This approach is popular because it provides accurate parameters with less computation and storage as compared to bank-of-filters and other approaches.

3.2.1.2 Recognition (Search and Match)

There are three technologies employed in recognizing the speech input viz. template matching, acoustic-phonetic recognition and stochaistic processing [Markowitz 96, p35].

Template matching is a form of pattern recognition; it matches an input utterance against a set of pre-computed templates (which are stored models). Inputs are compared as whole words or phrases: no reference to phonemes is considered. An input is compared to all the models and any similarity assessment is produced between the input and each model. The input will generally not match any model exactly, but the difference in matching is minimized by a process called temporal alignment, which is achieved through dynamic time warping. Template matching systems have thresholds of acceptability (the difference between data and the stored model), which serve to filter noise from data. If the template is compared with the models and no match exceeds the threshold of acceptability, it is assumed that no recording has been done and the system asks the user to repeat the input. Template matching systems perform well in small vocabulary applications but struggle when the vocabulary expands or when there are too many words that sound similar. For it to function, a template matching system must contain at least one template to which the input can be compared.

Acoustic-phonetic recognition works at phoneme level; the amount of acoustic data stored therefore depends on the number of phonemes contained in a language, rather than the number of words. As a starting point, the system examines the spectral patterns from the input in order to separate the phonemes from each other. It then applies some acoustic rules in which the extracted features are identified and boundaries clearly marked. Having done that, it will arrange the segmented and labelled phonemes into phoneme hypotheses. The hypotheses are matched with the existing vocabulary and the best match is taken.

Stochastic processing is currently the most dominant technology commercially. It has proved to be fast, efficient and robust. Like template matching, it requires storage of at least one model for each item that needs to be recognized by the system, but – as in acoustic-phonetic approaches – this is typically phoneme-based, rather than word-based. Stochastic-processing is non-deterministic and does not employ direct comparisons between the model and the input. The decision as to which model resembles the input is based on probabilistic analyses. The most popular stochastic approach employs Hidden Markov Models (HMMs) whose states and transitions contain information necessary to make decisions in the recognition process. The statistical information in the HMM states describes the parameter values and variances from sample words.

3.2.2 Vocabulary Representation

3.2.2.1 Word representation

Vocabulary is a measure of how many word templates a system can recognize. In the case of human beings, words are allocated meanings by employing two sources of information. The first source is the context in which the word is said and the second source is the prior knowledge that a person has with respect to that word. Speech-recognition systems store reference models of words to be recognized. These models are represented as templates, HMMs, sequences of phonemes or sequences of sub-words. The choice of the model representation defines not only the 'word' in the context of the system but also the recognition methods to be applied to the input signal. Any model representation chosen must address the variability in which words are spoken. The variability is mainly due to co-articulation, inter- and intra speaker differences. Below are some or the characteristics of the different model representations.

- Template

A template representation stores models as a sequence of vectors. Every vector contains a set of parameter values used in representing speech. This representation is simple and easy to generate and implement. No analysis is done on linguistic or acoustic-related information. In early systems, each template represented one spoken

instance of a word. This resulted in huge amounts of stored templates, especially when attempts were made to accommodate inter and intra-speaker differences. The representation therefore worked sufficiently for small vocabulary systems but produced huge storage and computation overheads in large-vocabulary systems, without achieving satisfactory accuracy. The solution employed was to create more 'robust' templates, by collecting more than one sample per word and averaging the acquired results using mathematical or statistical techniques.

- Hidden Markov Model (HMM)

The HMM representation is designed to capture pattern variations. The statistical information on the states and transitions are derived from multiple tokens of words from a user or sets of users. The nature of data in the states and transitions dictates the performance of the HMM, since training is invariably employed. The HMM is usually constructed in different structures but the most common structure is the left-to-right one. The HMM representation is fast, efficient, relatively accurate and very flexible. The argument against it is the fact that the statistical decisions on the current state are independent of the previous states.

- Acoustic-phonetic

The models in acoustic-phonetic representations are stored in the form of phoneme models. Comparing it to the previous two representations, it has an added advantage with respect to storage because spoken languages have a fixed number of phonemes and that number is fewer than the number of words. Therefore, an increase in vocabulary will not result in a corresponding increase in storage requirements for the system. This representation makes the systems ideal for inter-language portability. Developers employ different ways to create phoneme models. But the demand for large, generic and expandable vocabularies has forced developers to use computational approaches, such as neural networks and statistical approaches, which represent words in an efficient manner. Unfortunately, phoneme representations are often labour-intensive, as they require the creation of hand-labelled data.

- Subwords

Words have also been represented as subwords in the form of phones, Phone-like Units (PLUs), syllables, demi-syllables and diphones. The most successful subword representation is the so-called triphone, invented by Bolt Beranek and Newman [Markowitz 96 pp60]. A triphone consists of a phoneme (or PLU) surrounded by its left and right contextual information. Triphones are generally represented as HMMs with three states. The states contain statistical information for transitions from the preceding phoneme to the current phoneme, for the current phoneme and for transition from the current to the next phoneme. This statistical information includes variability because of co-articulation, the effect of which is catered for using triphone models. The incorporation of the co-articulation information results in multiple triphones for each phoneme. HMM triphones are stored in databases and are concatenated to form words. The construction of this representation involves collecting samples of spoken tokens from the users, breaking the tokens into phonemes (organized by triphones) and the representation of words using these phonemes. The built words are then stored in a database with pointers to the relevant phonemes out of which they are built.

3.2.2.2 Word identification

When represented in a vocabulary, words are assigned unique tags in order to distinguish them from one another. The most common systems use printed word representation as identifiers. Ten, for example, can be represented as ‘10’ or ‘TEN’, that is, using the same characters strictly in that order. The use of printed word representation makes it easy to differentiate between homophones (e.g. one and won). Some systems employ case sensitivity, in which case, “One” and “one” will represent two different words. There has also been a growing trend to represent short, common sentences or phrases as one word by using hyphens. A good example is the phrase “end of application” which can be converted into a single word “end-of-application”.

3.2.2.3 Word translation

The main function of an SR system is to convert the speech input signal into the form that the underlying applications can use. Depending on the application that utilizes the input, the signals can be converted to keyboard strokes, touch tone pulses or

even to paragraphs of text. Many SR systems do not attach meanings to the words: the words are translated into their equivalent, say, graphemes. Speech patterns in SR must have unique translations at any specific point in an application, to avoid ambiguity, but multiple translations within the same application are allowed, provided that these cannot occur at the same point.

3.2.2.4 Word variants

Word variants are represented as different entities in the vocabularies, (unlike in conventional dictionaries, where a word is often grouped with its conjugations, derivations, etc.). For example, *do, did, done* and *doing* are represented as four separate unrelated entities.

3.2.2.5 Vocabulary design.

Words are usually stored in vocabularies in terms of subwords, rather than with distinct acoustic representations (see above). To construct say, a 50 000 word vocabulary requires the processing of significant quantities of text, and most developers rely only on online-based reference databases for the supply of relevant words. However, reference databases are often too general, may contain misspelled and foreign words, proper names and unpronounced or wrongly pronounced words. They must therefore be tailored for specific applications. Even with all these adjustments, there is no guarantee that all the required words for an application will be contained in the chosen reference database. The system may therefore be required to detect out-of-vocabulary words. Current systems detect this when either no recognition above the threshold of acceptability has been recorded or an erroneous word has been chosen. Systems differ in how they handle out-of-vocabulary words. In many cases, the user is provided with a list of words to choose from. If no choice is made from the list, then a new model is created. Vocabularies are provided either by the vendors, by the application developers, by end-users, or by any combination of the above.

Vendor-supplied lexicons are usually found in large-vocabulary systems, turnkey applications, systems embedded in firmware and speaker-dependent user model systems. The whole vocabulary is created by the vendor and not by the application

developer or user. They are created in the form of dictionaries or application-specific vocabularies. Dictionaries are called total vocabularies and contain all coded words (sometimes up to 100 000 words). They are usually sourced from online or printed documentation. Application-specific vocabularies are tuned to the specific application. Such dictionaries are generally able to provide enhanced recognition accuracy, at the cost of significantly increased application development time. They are imperative for systems to be used by naïve and disabled users. When used in large vocabulary systems, they represent a subset of the total vocabulary of the system and are sometimes called resident vocabularies.

Users have their unique requirements. They can therefore not always rely on the set of words that vendors or application developers will come up with. Developers and vendors can generally not predict items such as names and acronyms. It is therefore necessary for provisions to be made to enable users to expand or even to create their own vocabularies. New words also need to be accommodated without taking the system back to the application developer or vendor. In these circumstances, vocabulary development tools must be provided for the users to create or expand their vocabularies.

For vocabularies that are extracted automatically, the systems extract the words from an online file or system. This is sometimes called application scanning or vocabulary optimisation. The whole task is done by the system and the vendor is therefore freed from it. This method offers personalization of large-vocabulary systems, and caters for specific needs. It is therefore bound to grow in the foreseeable future. However, such vocabularies do not work for offline applications.

3.2.3 Structuring the Vocabulary

Vocabulary is stored in a structured manner in order to reduce perplexity. Reduced perplexity results in increased recognition speed and accuracy. What is perplexity?

The number of choices for a particular search is called the *branching factor*, and has a significant impact on recognition accuracy. The average branching factor in an application is called the perplexity. For example, a ‘yes-no’ system has a perplexity of

two. Business systems have perplexity ranging from one to some hundreds. Reducing the perplexity decreases the complexity of the searching process. When the active vocabulary is structured, fewer comparisons are required in the recognition process. This will increase the speed of the recognition system and will also reduce the number of errors. All these will in turn result in an increased usability for the whole system. With arranged vocabulary, confusable words and homophones are placed in different active vocabularies. This arrangement will minimize errors caused by these words. Vocabularies are structured using, Finite State Grammars, Statistical Models, Linguistics-Based Grammars or Word Spotting. The four methods are discussed briefly below.

3.2.3.1 Finite State Grammar (FSG)

An FSG reduces perplexity by excluding some words at any point. It works by replacing a list of active vocabulary with generic descriptions, e.g. distance (*NUMBER*), where NUMBER must be substituted with a numerical value. Descriptions can then be embedded in other structures to make longer sentences. FSGs are usually stored in a form of state networks. Each state is linked to its left and its right. The states and transitions are not linked according to probabilistic information (as in an HMM), but the networks are larger and can typically model complete sentences.

The FSGs are relatively inflexible: since it can only recognize utterances that fit the descriptions, the input is restricted to a rigid set of predefined conditions. These grammars do not consider language syntax or statistical information. All words in the active vocabulary have equal probabilities, i.e. they are not ranked according to their probabilities of appearance. This method is straightforward, easy to understand and simple to create and implement.

Finite State Grammars are suitable for structured applications (data entry, voice command and control etc.). They are fast, efficient and relatively accurate in comparison to the other models [Markowitz 96 pp81]. The challenge when in use is to make sure that the user speaks only the allowable words. It is this challenge that

makes them more suitable for smaller groups of users who are to be trained (otherwise, the accuracy of the system will be too low).

3.2.3.2 Statistical Models (N-gram models)

N-gram models are the most common statistical models in use both in commercial and research systems. They identify the probability of a word using probabilities from the previous N-1 words (as well as the acoustic information from the target word). The active vocabulary is usually smaller than the entire vocabulary. This model is suitable for large vocabulary applications as it improves and optimises its performance by continuously updating the probabilistic information as the system is being used.

N-gram models extract useful sequence patterns from large quantities of data. Once the patterns are collected, the system is 'trained' by ranking and coding the possible nth words according to their probabilities of occurrence. The result of this ranking and coding is an N-gram model, which is typically represented by lattices or linked lists. During application, these links are used to select, rank or eliminate words. The amount of data required to create a good N-gram model is a function of both the vocabulary size and the value of N. N-grams can be optimised for personal preference by adding personal words or by adjusting the grammar to suit a particular user. N-class (Biclass, Triclass etc), which extends the N-gram idea to semantics and/or syntax, is a special case of N-grams. A biclass for example gives the probabilities that two classes of words will follow each other.

A major weakness of N-grams is that it has a limited window. As an example, a tri-gram will only consider the information it gets from the previous two words to determine the third word. Even when a sentence of ten words is used, it only uses the information based on the previous two words. N-grams (bi-grams and tri-grams in particular) are suitable for large vocabulary systems. However, they are difficult to construct and require large amounts of data and sophisticated statistical analysis. The best sources of data are organizational documents where relevant words can be extracted for the system's vocabulary.

3.2.3.3 Linguistic-based grammar

The idea behind the linguistic based grammar is to both hear and understand what was said. Knowing what the user wants can refine the active vocabulary and help improve the speed and accuracy. Linguistic based grammars are mainly implemented in the form of context-free grammars, which are implemented alone or as part of a combination with other grammars. Just like finite state grammars, they define allowable structures. They are able to represent a variety of linguistic approaches, are powerful, flexible and have simple representation. On implementation, the context-free rules are applied to the speech input and the rules that are successful are then placed on a chart, which represents both recognized words and their meaning.

3.2.3.4 Word Spotting

Word spotting works by identifying a set of predefined target words from a stream of input. Target words are stored as models in the form of HMMs or templates. Due to factors such as noisy environments and user variability, a word spotting system may fail to spot valid target words (false rejections) or may spot words that were not actually said (false alarms). But these errors are minimized by selecting non-confusable words, by using appropriate non-keyword word models or by filtering speech properly from background or channel noise. The information to be extracted is normally defined by type (e.g. ID number, PIN code etc) rather than by grammar.

Word spotting is mainly used in small, very structured applications in which individual users vary largely. It is difficult to implement with accuracy because it has a greater number of potential users, the users are not committed to the system and the nature of the communications to which it is applied is usually transient. Word spotting is usually invisible to the targeted users and it does not interfere with the conversations.

3.2.4 Speaker Modelling

Speech-recognition systems are supposed to cater for different sets of potential users. Recognition is achieved when incoming speech is compared to stored patterns and a satisfactory match is found. Speaker models in use are speaker-dependent models, multi-speaker modelling, speaker adaptation and speaker-independent

models. The four models will be discussed below, with more emphasis on speaker-independent and speaker-adaptative models as the two most popular approaches.

3.2.4.1 Speaker-dependent models

The speaker-dependent approach creates a model for each user who is expected to use the system. Each model contains speech attributes for that particular user and is stored as a separate file. For any user to start using the system, the model that contains her attributes must be loaded first. Creating a user model entails data collection, calculation and model construction. Data collection, or training as it is popularly know, mainly deals with eliciting token(s) of spoken words from the user. Calculation and model-construction details differ, depending on the model employed (HMM or templates).

Templates: Norms for acoustic parameters are calculated within the collected tokens. Templates are then constructed using the computed results. Early systems used to calculate a template for each token, which resulted in multiple tokens for each vocabulary item. The process, as expected, consumed substantial storage resources and required numerous comparisons and therefore much computation. It was limited to small vocabulary systems and could not handle intra-speaker differences properly. As an improvement, developers started to employ robust training, where acoustic data from different tokens are averaged and one template is formed using that average.

HMM: Calculation utilizes more sophisticated statistics, not just averaging as in templates. The statistics capture the variability and norms of users. Results are embedded in the states and transitions of the Hidden Markov Models of words. The Baum-Welch algorithm is commonly used, which recalculates and updates the tokens based on the newly received data. The process of recalculating and updating is called re-estimation.

3.2.4.2 Multi-Speaker modelling

Multi-speaker modelling extends speaker-dependent modelling to multiple users. Models are designed to represent attributes of a group of users rather than just one user. Its implementation requires the knowledge of the user population, so that each user supplies at least one token per sample during data collection. Systems using this

modelling are less accurate than speaker-dependent systems, since the averaged parameters hardly resemble any of the tokens supplied by individual users.

Multi-speaker modelling is suitable for small to medium vocabulary systems where loading and unloading of models are not possible and/or desirable, more especially where the application alternates rapidly between users. It is not suitable for security systems as the acoustic patterns for users may be similar. The accuracy of these systems is better if the population is small and homogeneous, but it degrades with the variability of the user population. Addition of new words after deployment is very difficult because tokens from all sampled users are required.

3.2.4.3 Speaker-independent (SI) models

Speaker-independent modelling is designed for use by more than one user without enrolment. Its implementation is made difficult by individual differences (especially factors such as speech impediments) and different accents of the same language. The models represent the expected acoustic parameters of the entire population. They are difficult to construct because the inter-speaker differences that have to be catered for, are usually unknown at the time of construction. SI systems have been most successful in applications with small-to-medium sized vocabularies. Reference models are created by sampling, subword modelling or neural networking.

Sampling follows the steps in speaker-dependent modelling i.e. data collection, calculation and model creation. Data collection is more complex and extended than in speaker dependent modelling. The number of tokens required depends mainly on the population size, the vocabulary type, the speech flow and the speaker environment. For a very diverse user population, more tokens are required. The population is typically divided into representative subgroups and tokens from each subgroup are used.

If templates are used, then acoustic data are clustered into groups of similar patterns. Common patterns are then extracted and the centre for each is taken as the template for that pronunciation.

If *HMMs* are used, all acoustic data are collected and calculations are done using the Baum-Welch or other relevant algorithms. Thereafter, a single set of HMMs is constructed for all speakers. Sampling produces good speaker-independent systems but they are difficult and labour-intensive to implement.

Subword modelling is suitable for large vocabularies and is also used for speaker adaptation (discussed below). The idea behind subword modelling is to construct a database of subwords, develop a technique to concatenate them into words and define the form of user input. The database constructed will host all the subwords to be used on the system.

Subwords are stored as statistical models, e.g. as HMMs. The data is taken from spoken language sources (both generic and application-specific sources are widely used). After collection, data is segmented into subwords. Calculation and model construction is done as in sampling (discussed above). Results of the calculations are tested and stored in a database.

Neural networks in speaker-independent systems take a different approach. The most common approach is to create a prediction network using a feed forward architecture. The networks are trained to predict the subword produced, based on the acoustic parameters of the input. Some systems group speakers into subgroups represented by clusters of reference speakers. Each speaker is then considered as an instance of a subgroup. As soon as that is done, every input is compared with the reference of the selected subgroup (cluster) [Markowitz 96 pp109-110].

Speaker-independent recognition is suitable for one-time, medium-vocabulary applications (for example, telephone and kiosk-based applications). Accuracy is far less than in both speaker-dependent and multi-speaker modelling, due to the variability of user groups. Models are created through sampling or subword modelling.

If models are created using the *sampling method*, the performance of the systems depends on the number of samples used per model, the representativity of the samples in terms of the user population and the deployment environment as well as

on the quality of the algorithm used to generate the models. The number of tokens per model depends on the size and diversity of the user population, the noisiness of the environment, the accuracy requirement of the system and the characteristics of the vocabulary. Heterogeneous populations require more tokens per vocabulary entity than their homogeneous counterparts. However, it must be stressed that speaker-independent systems are not 'one size fits all' systems. They must still be designed with a definite target user population in mind. A change in either the population or the environment may decrease the degree of accuracy. Noisy or changing environments as well as confusable words need more samples. Multi-syllable word entries need fewer tokens than one-syllable words, which are more difficult to recognize. Sampled models are also vulnerable to the *recitation effect*. The most common method to address this effect is to collect data while the system is in use.

Creating models using *Subwords* is relatively easy, fast and inexpensive in comparison to the sampling method. But it has drawbacks in the acoustic representations (words are divided) and environmental changes. Therefore, some operational systems in this category rely on specific samples to handle critical and confusable words.

Speaker-independent systems contain vocabularies constructed from a large number of samples or by melding of subwords. Almost all vendors offer vocabulary development tools. If the system is meant to be used by a single user or a small group of users, then the vocabulary can be created by the users themselves. Both sampling and subword modelling can be created by application developers. If there happens to be a change in the dialect, speaker environment or speech channel, the models need to be adjusted. Most vendors offer modification tools to that effect. In particular, sampling based systems are offered with modification tools while subword based are modified by adding new data.

3.2.4.4 Speaker Adaptation

The idea behind speaker adaptation is to modify existing models to suit new circumstances. This enables the speech recognition system to process inputs that are

different from those used to create the models. It requires a small amount of data initially, since it utilizes data efficiently.

Speaker conversion entails shifting speaker-dependent reference models for one user to reference models of the other. This is usually done at the word level.

Enrolment adaptation takes data from the speaker-independent system and transforms them into data for one speaker (as if transforming it into a speaker-dependent system).

On-the-fly adaptation modifies the reference models after the system has already been deployed. The aim is to enhance user acceptance while avoiding costly user enrolment. In order to maximize accuracy, it can be used after the enrolment has been completed. As in the other modelling, adaptation is applied to templates, to HMMs or to neural networks.

Template adaptation is not very common. It is performed on the whole word and templates must be created using data from robust training. Adaptation is done using a small amount of data from a user to modify the existing models. By enrolment or on-the-fly adaptation, data from users is clustered with data from the templates.

HMM adaptation is sometimes applied to words but it is commonly applied to subwords in large-vocabulary systems. Data from the user is used to modify subwords. The modified version is stored in a file representing a single user. Two common representation approaches are employed: vector quantization codebook adaptation and the HMM acoustic parameter adaptation. Vector quantization codebook adaptation modifies code words in the vector codebook. The other approach, HMM acoustic parameter adaptation, modifies the acoustic parameters of the HMM itself so that it resembles data from a specific speaker.

Neural Networks for speaker adaptation neither modify internal model structures nor create permanent models for new speakers. Any user is classified as a member of one of the existing clusters, just like in speaker-independent systems. It is usually implemented in hybrids where each node in the network provides probabilistic characteristics on the input. For example, speaker-adaptive neural networks were implemented where the input is first characterized as being from a male or female, then put into one of the many clusters [Markowitz 96 pp113].

Speaker adaptation systems are suitable for large vocabulary systems with repeated usage where enrolment is deemed to be cumbersome. They are not good for one-time users. Accuracy is improved by adapting the model with every major change that occurs. Accuracy can be low at the beginning of use, but it will improve as the system collects more data during usage. The addition of new words is achieved in a manner depending on the technology used to create models. Large vocabulary systems normally contain backup dictionaries and new words can only be added by vendors or by subword modelling technicians.

3.2.4.5 The Speakers

Stress is one of the major causes of decline in accuracy of the speech recognition systems. When a user is under stress, he/she tends to speak differently. Stress can be caused by a number of factors, including poor system performance. If possible some tokens must be collected from users while under stress. It must be noted that when too many users' input are being rejected by the system, it is likely that the problem may be with the system and not necessarily with the users. It is also of utmost importance that the users accept the system. System acceptance is affected by the perceived benefit of the system, participation of users in the development of the system, the fear of technology as well as by the fear of change.

3.2.5 Flow of speech input

Ideally, users are supposed to communicate with the computer system in a natural way. However, they may expect the computer system to have unlimited vocabulary and the interface to have a very flexible grammar. Unfortunately, the existing technology cannot offer all these capabilities yet. Speech from the user can be categorized in terms of the flow of words as perceived by the system. The most prominent categories are *discrete words, connected words* and *continuous speech.*

In discrete-word applications, the user must pause between every pair of words. This will prevent the acoustic information from being distorted by co-articulation and provides time for the system to process the most recent input before proceeding. The

pause between words has to be at least 15 milliseconds in typical systems. Discrete-word input is used as input into either template models or HMMs. When HMMs are used, boundaries between models consist of silence or background noise, which facilitates recognition of word boundaries. Discrete words invariably enhance recognition accuracy, but users generally find it very cumbersome to speak in discrete words.

Connected word (popularly known as connected speech) input works more or less like discreet word input but the pauses between words are shorter.

With Continuous speech input, the speaker is supposed to speak naturally without enforced pauses or hesitations between words. Although speech-recognition systems have long been designed with this goal in mind, its achievement has been hampered by difficulty in locating word boundaries and the effect of co-articulation. Unlike discrete recognition, where word boundaries are obvious, word boundaries must be located in continuous speech. In locating the word boundaries, care should be taken to limit the search to a few possible candidates otherwise the process will result in very high computational costs. In many cases, heuristic techniques (besides direct comparisons) are utilized.

Cross-word co-articulation results in the alteration or deletion of individual phonemes. It strongly affects one-syllable words. In small-vocabulary systems, this problem is handled at word level, using structuring techniques (such as grammars). Word pairs or finite state grammars help in identifying the next word and restrict the active vocabulary. In large-vocabulary systems, co-articulation is handled at the subword/triphone level.

The other problem with continuous speech recognition is the design of appropriate sentence grammars. Natural speech contains a wide variety of constructs, and the development of grammars to capture this diversity is, in general, an unsolved problem. This explains the logic behind connected-word recognition, where grammars are restricted to be concatenations of items from a very limited vocabulary.

3.2.6 Speaking environment

Speech recognition systems are expected to perform accurately in the environment in which the target application is deployed. The environment can vary from a very quiet office to a very noisy factory or even a mobile telephone in a busy public space. For this reason, the ability of the speech recognition system to perform under diverse conditions becomes a critical issue to the success of the system. The major problem within the working environment is the noise. Noise is defined as the sound that is transmitted into the system, but which does not form part of the information conveyed by the user. There are four kinds of noise associated with the operations of a speech recognition system. They are background noise, channel noise, speech noise and non-communication vocalization.

Background noise refers to the sounds that are produced around the place where the speaker is using the system (excluding the user speech itself). Channel noise is the part of the sounds that are added to the speech input by the carrier while transmitting it from the speaker to the application. Speech noise and non-communication vocalization refers to the speakers' contribution to the noise, which include clicks of the tongue and expressive phrases like 'uuuuh'. Speech input is regarded as 'clean' if it does not contain any noise, otherwise it is called noisy or corrupted speech. Each of the noises contributes to the difficulty in the recognition process in its own way. The speaker environment characteristics that greatly affect the performance of a recognition system are: the nature of the background noise, variability in the noise, loudness of the noise and the type as well as the quality of the speech channel in use.

3.2.6.1 Signal-to-noise ratio (SNR)

Signal-to-noise ratio is a measure of the speech/information component as compared to the noise/unwanted component in the same input signal. It is measured in terms of the response of the input device (microphone) to the background noise and other speaker environments and is measured in decibels (dB). A higher SNR number implies a stronger speech signal compared to the noise signal, and vice versa.

Recognition accuracy is monotonous with the SNR. Most natural environments have low SNR and therefore the analysis of SNR must be done properly for the recognition systems to achieve their performance requirements.

3.2.6.2 Background noise

Background noise is the sound produced around the speaker, which finds its way to the microphone and mixes with the required input signal. It is produced by machinery, by fellow people, by passing vehicles, by phone rings or other sources of noise in the surrounds of the speaker. Background noise cannot be completely eliminated, and it is therefore important to minimize its effects on the recognition process. Background noise is described in terms of the SNR and the variability of power spectrum. Variability is the rate of change of patterns in the noise with respect to time. Monotonous noises, for example noises from running machinery are less problematic than intermittent noises like ringing of a telephone or knocks at the door. Highly variable and intermittent noises are unpredictable and are difficult to model and handle. The power spectrum is the frequency range within which a certain sound exhibits great intensity. If the power spectrum of the noise co-insides with that of the speech input, noise reduction becomes extremely difficult. However, the rate of change of acoustic patterns for most background noises is much lower than that of speech input. This makes it easier to filter out the background noise. But if background noise contains human voices (radio, people talking, etc.), filtering of background noise becomes significantly more complicated.

Another important background noise is reverberation, which is usually produced as a result of an enclosed environment. The resonating sound is altered by the enclosure, where some frequencies are amplified and others are reduced. Reverberation is affected by the shape and size of the enclosure, construction material used, the presence of other bodies (e.g. furniture), the type of the microphone used and the speaker's distance from the microphone. Another major problem with reverberation is that it repeatedly feeds the same information into the microphone and hence into the system. In practice, reverberant noise is most important if speech originates from a speakerphone.

3.2.6.3 Channel noise

Channel noise is the noise added onto speech input by the capturing and carrier devices (microphones and telephone network, respectively). These channels transform analogue sound waves into digital format, and transport them to the recognition system. In the process they also introduce some forms of distortion as well as electrical noise. The effects of channel noise cannot be ignored. It has been shown that, in some speaker-independent systems, the effect of channel noise exceeds that of speaker variability.

3.2.6.4 Microphones

Each make and model of a microphone introduces its unique noise and distortion. These effects are usually summarized in the manual of the microphone. It is therefore important to use the microphone that is recommended by the developer, whose noise and distortion characteristics are known. There are also differences in types and qualities of microphones. Omni-directional microphones have uniform reception patterns in all directions, while the reception of directional microphones depends strongly on the direction in which they are facing. Directional microphones are therefore good at filtering background noise as they can be turned into different directions as desired.

Noise cancellation microphones have the ability to filter noise from specific sources. They are constructed from a pair of bi-directional ones and are used with one facing the speaker and the other facing the sources of noise. The microphone that faces the source of noise will capture the patterns of noise, which can be removed from the speech input before transmission. Close-talking microphones are meant to be placed as close to the input source (speaker's mouth) as possible. They are insensitive to the sounds produced farther from it and therefore minimize the effect of background noise and reverberation.

If the system is designed to use telephone lines, it must be able to handle handset and line noise and distortion. The types of microphones used for telephone-based systems are: *the electret speakerphone*, *the carbon button handset* and *the electret*

handset [Markowitz 96, pp 153]. They all differ in the reverberation they capture and the distortion they introduce to the input speech.

Systematic network noise refers to a permanent feature of the network. It is the overall amount of noise introduced by all the network equipments. This noise is mainly consistent, predictable and known beforehand, although it varies from one network to the other. Random network noise is mainly due to cross-talk crackles and atmospheric noise. It is usually introduced at irregular intervals and is very unpredictable. If communication is done within the same network, the cannel noise may be determined, but with inter-network communication, it is very difficult if not impossible to determine the channel noise. Wireless-based speech recognition introduces additional challenges, as the transmission environment varies greatly. As handsets are mobile, it is not possible to predict with accuracy the background noise of the user/microphone at all times.

Non-communication speech noise refers to the non-information sounds that are produced by the speaker while producing the speech input. Examples of these are: the smacking of lips, air puffing, throat clearing and tongue clicking. A speaker may yell or exclaim with words like “uuuuh”. She may also make verbal corrections like “It is in Johannesbu... no, in Pret..yes Pretoria”, instead of a single and correct input “Pretoria”. Some of these noises are predictable and can be handled by the recognition system as garbage. But most of these sounds are hard to predict. Successful modelling of these will be aided by the knowledge of the speaker goals, conversational structures and other cognitive-linguistic resources.

3.2.6.5 Speaker response to background noise

Users are affected by the background noise and they react accordingly. When confronted with such noise, a user may increase vocal efforts (shouts), increase the duration of words; shift the formant location of words; increase certain formant amplitudes, or delete some word-final consonants. All these changes are jointly known as the Lombard effect. The Lombard effect aids in human-human communication but it decreases the effectiveness of a speech-recognition systems. The nature and consistency of the acoustic changes due to this effect are not globally

consistent, and existing speech-recognition systems do not correct for the Lombard effect.

Before attempting to handle the noise, it is important to know the type and characteristics of the noise in the application's environment. To know this, an assessment has to be carried out in all the environments where the application will be working. Parameters assessed in the analysis of noise include:

- The nature of background noise
- The variability of the noise
- The loudness of the noise
- The type of speech channel used and
- The quality of the channels used.

3.2.6.6 Techniques For Designing Robust Speech Systems

When speech recognition systems were confined to offices, there was hardly any reason to worry about enabling the systems to perform in different environments. But as the usage of the speech recognition systems spread to other environments, robustness became very important. Many attempts to improve on robustness have been aimed at minimizing the effect of additive background noise. The characteristics of the noise are captured by measuring the noise contents during non-speaking periods. When the characteristics are known, its effect on the systems can be reduced. Below are some of the methods used [Markowitz 96 pp161-172].

- Speech enhancement

Speech enhancement is an attempt to improve the quality of the speech itself rather than filtering it from the garbage. The two techniques, noise reduction and speech enhancement, together with the selection and placement of the microphone can make a big difference on the robustness and accuracy of the system. Nevertheless, no current speech recognition system can handle noise with anything approaching the capabilities of humans. Therefore, application developers must be made aware that the presence of noise within speech input will continue to reduce recognition accuracy.

- Training

If the noise is known in advance, multi-context training must be employed, where the training is carried out both in quiet and in noisy environments. This approach works well in environments with consistent background noise. But if the noise is inconsistent and unpredictable the best approach is to try and develop some immunity within the model itself. Some research suggests that adding noise to a system designed for a quiet environment during training improves its accuracy, but this approach is still fairly controversial. Others therefore attempt to improve the algorithms that generate the models, but the degree of success to date has not been encouraging.

- Pre-processing Techniques

Pre-processing is the process of removing the noise from the speech input before digitisation. There are a number of techniques employed to carry out that process; the simplest approach is *band-pass filtering*. Filtering works by restricting the frequencies of sounds processed by the system. Filtering can be categorized as low-pass, band-pass or high-pass filtering, depending on which frequencies are accepted. Band-pass filtering allows frequencies in a certain range to pass through and has proved to be useful in practice. Filters are based on the linear scale (in Hz) or the Mel scale.

- Noise cancellation

The noise cancellation process removes the frequencies associated with noise from the speech input. The common techniques are spectral subtraction and masking. With subtraction, the frequency spectrum of the noise is estimated when there appears to be no speech input (based on the signal amplitude). The noise spectrum is then subtracted from the signal received. This process works best if the frequency of the noise is quite distinct from that of the speech input. If the frequency of the noise co-insides with the frequency of the speech input, masking is generally applied. Masking simply eliminates all frequencies that are expected to be dominated by noise.

- Handling of the channel noise

Even the best quality microphone may under-perform on some application environments. Most applications require high quality, bi-directional phones. Some vendors recommend certain brands and types of phones, but it is always advisable to use the same type and model of the phone that was used during enrolment.

- Handling of non-communication speech

The presence of non-communication speech can create problems in the speech recognition process, but handling it is much more difficult than handling background noise as its spectral characteristics are usually similar to that of the informative input. Some non-communication speech can be handled by the current systems with the proper placement of the microphone, so as to eliminate speech not intended for the speech-recognition system. Well-designed applications, good reference models and proper training of users also minimize the problem of non-communication speech. If possible, reference models for frequently used noise words must be created to aid in filtering them from the speech input. Backup and other error correction mechanisms can also be used to recover from errors. Good design of speech aware applications will also help the recognition systems to keep synchronization with other computer system components.

- Handling of the Lombard effect

Little attention has been given to addressing the Lombard effect on speech recognition processes. The most prominent technique utilized has been Multi Style Training, but this technique works well only if the vocabulary is small, if the noise in the working environment is uniform and if the users are cooperative. In many cases, the Lombard effect characteristics changes with the change in the noise conditions. This remains a topic for active research.

3.2.7. Speech Recognition Systems: Performance and accuracy

The performance of speech-recognition systems is affected by the environmental noise in which they operate, the users' speaking style, the diversity in the user population and the complexity of the application. In the 1980s, speech-recognition systems were confined to quiet rooms, to discrete-word input, to specific speakers and to a small set of applications with very limited vocabularies. Designers and

developers alike are now striving for systems that serve any operation environment, any speaking style, any application and systems that can be utilized by any user. Figure 3.1 below [Picone 00] gives a summary of the complexities that effect SR systems' performance.

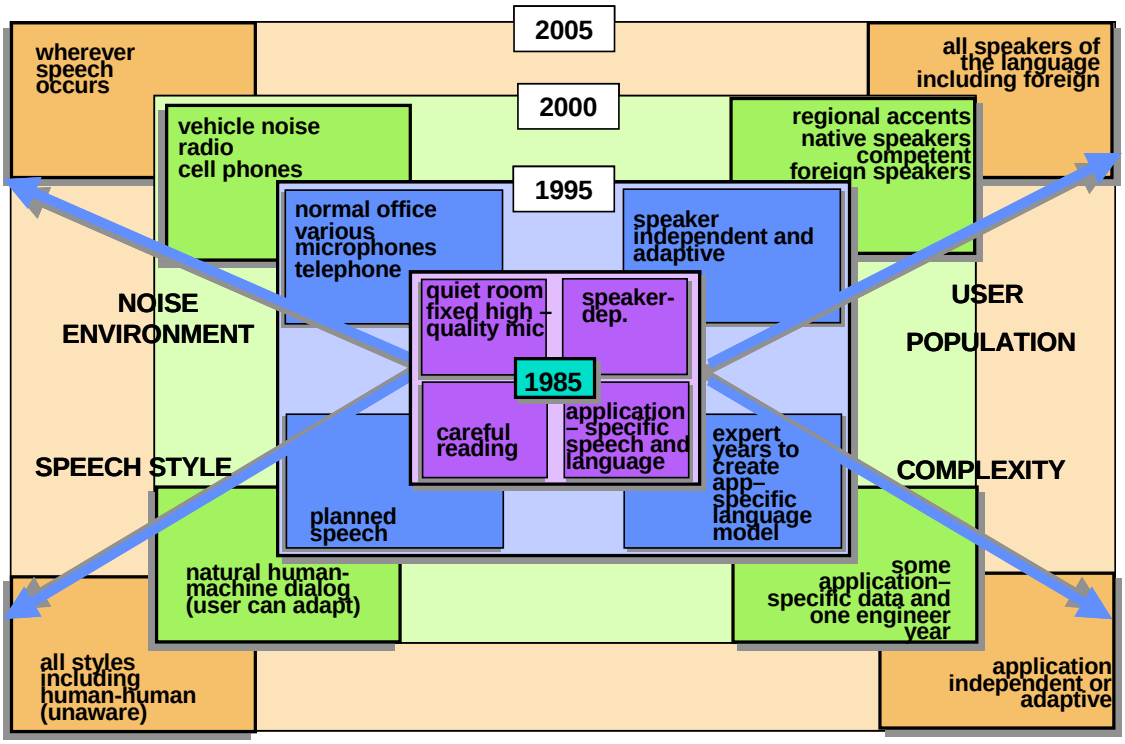

Figure 3.1: The current State of Art in the Performance of SR Systems

Two major measures can be used to evaluate the utilization of speech-recognition applications: accuracy and transaction-completion rates. Accuracy is a measure of how many words were recognized correctly against the total number of spoken words. Transaction completion is the number of transactions that were eventually completed against the total number of attempts. If the recognition accuracy is very high, it is likely that the number of transaction completed will also be high, although not always. But if the accuracy is low, it is likely that the number of transactions completed will also be low. Again, low accuracy does not necessarily imply low transaction completion rate, more especially if there are alternatives to the use of speech recognition systems.

When a manufacturer of a car claims that a car can reach a maximum speed of 200km/h, it is always assumed that the testing was done on a tarred road, unless

otherwise stated. Speech recognition systems do not have tarred roads to run on. Therefore when a designer claims that the speech recognition system has an accuracy of, say, 98%, it must be put into context. He has to specify the size of the vocabulary involved, the speech flow, the level of the background noise and the signal strength. The aim of structuring the vocabulary, modelling the speaker, the speech flow and the speaking environment is to get the best accuracy possible. But even if we do that properly, recognition accuracy is also highly affected by the signal strength and the composition of the vocabulary.

3.2.7.1 State of Art in Recognition Accuracy

Figure 3.2 below gives information on the current state in accuracy of the speech recognition systems with respect to the size of the vocabulary and the way the speech input is fed onto the system [Picone 00]. It shows the error rates achieved on various tasks during the past 15 years.

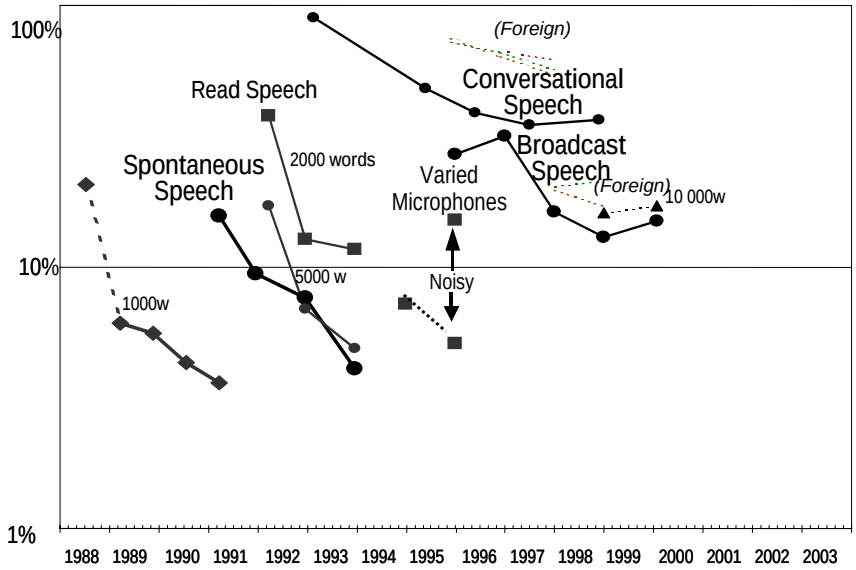

Figure 4: State of Art in the Recognition Accuracy

While recognition accuracy in general has improved drastically over the years, more challenges have emerged.

An accuracy above 90% (error rate below 10%) is generally considered acceptable for command-and-control applications.

These measurements are lab based and accuracies may decrease by a factor of 2-4 in the field.

3.2.7.2 Signal strength

The difference between the amplitude of the noise and that of the speech input greatly affects the recognition accuracy. If the signal's amplitude is a factor of, say, 10

(to that of noise) then the accuracy will be higher than if the factor is 2. And if the amplitude of the noise is larger than the amplitude of the speech input, the recognition accuracy will definitely be unacceptable. It is therefore appropriate to specify both the noise level and the signal strength when quoting an accuracy of a speech recognition system. Particularly, it is recommended to state both the optimal (the lowest noise level from where reducing the noise will not affect the accuracy any more) and the highest (the highest level at which the system will still perform acceptably) with respect to a given signal strength.

3.2.7.3 Fluctuation of the noise level

Fluctuation of the noise level decreases the accuracy. If all other factors are kept constant, a system operating at 50dB of continuous but constant noise will yield higher accuracy than the same system operating at an average of 50dB of highly fluctuation noise. Therefore if possible, the fluctuation in the noise level must be specified when quoting the speech recognition accuracy.

3.2.7.4 Size and content of the vocabulary

Besides the structure of the vocabulary, the size and content of the vocabulary also affects the accuracy. The smaller the size of the vocabulary, the more accurate is the system, at least in theory. A system with accuracy of 96% on a 2000 word system is likely to have less accuracy on a 20 000 word vocabulary. The number of sets of similar words in the vocabulary also decreases the accuracy. The system is likely to confuse whether it heard "*Duke Gordon*" or "*Juke Korden*" or whether it heard "*wreck a nice beach*" or: "*recognize speech*". All other factors aside, the greater the number of confusable words, the less is the expected accuracy of the system.

3.2.7.5 SR systems accuracy compared to human

How close to human recognition accuracy are current speech-recognition systems? An ordinary user expects a speech recognition system to recognize speech accurately as a human does. But speech-recognition systems are far from attaining the human recognition accuracy. Figure 3.3 [Picone 00] below shows the results of comparing state-of-the-art SR system and humans in terms of word error rate.

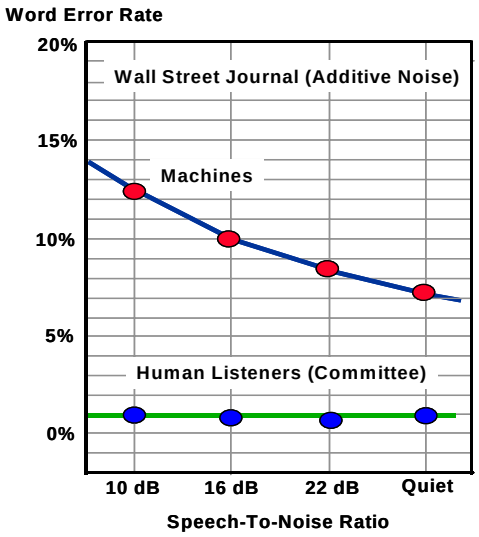

Figure 3.3 Comparison of Human and System error rates, at various noise levels

The human error rate is consistently lower than that of the speech recognition system - up to a factor of 10 at low signal-to-noise ratios.

Where there is memorization and memory retrieval, the system may out-perform the human

The human performed consistently over a wide range of background noise while the system's performance is highly affected by the speech-to-noise ratio.

3.2.8. Speech recognition applications

Speech recognition systems are used as input interfaces for data entry, command and control, information access or dictation applications. Successful utilization of speech recognition requires a thorough understanding of the capability on the existing technology, the requirements of the application and the users who will be using the system.

3.2.8.1. Human factors

The technology issues have been dealt with in the previous sections. This section will concentrate on understanding the human factors, which includes understanding the users as well as the task they are to fulfil.

Who will use the system that is being developed? That is the main question to be answered if a thorough understanding of the users is to be achieved. Users differ in their personal characteristics, task related attributes and their abilities. If the speech recognition system is to be used on an existing application, a greater part of the vocabulary for experienced users may be in existence already. It just needs to be extracted from the relevant sources and be compiled. If the system is for novice users, they may prefer to use a natural-language vocabulary, as they are not used to

the terminology of the application. This means that the system may contain two or more sets of users. Experienced users may need barge-in facilities as they are well acquainted with the procedures and flow of the application's tasks. Unlike experts, novices may need guidelines and lengthy explanations at every step of the process. Speech-recognition systems perform better than other interfaces for people with little or no knowledge of computers. But it must be borne in mind that some people are scared of new technologies including computers and the interfaces thereof.

Why do people want to use the system? Users normally have well-defined goals ranging from getting information, placing a call or even dictating a document. It is important that the users are made to believe that using the speech system will make them fulfil their tasks more easily, more quickly and with a higher-quality end result. The users may object to some facets of using speech systems, for example they may refuse to wear headsets or even totally refuse to 'speak' to a machine. In all circumstances users' needs must be satisfied as far as possible. While speech recognition systems were confined to a few applications, the usage and user population has expanded in the commercial sector. That has resulted in greater horizon and wider range of human factors to be considered.

The objectives of the tasks are normally reflected in the structure of the system, the organization of subtasks, on the input modalities and on the communications it makes with the users. The goals of the tasks define the elements of the human interface, including: the information obtained from and given to the users, the options available to the users, positioning of input/output devices with respect to the users, interface modality available to the users and error handling methods.

3.2.8.2 Examples of Applications that use speech recognition systems

As mentioned earlier, speech recognition is widely used for applications for command and control, data entry, information retrieval and dictation. The four categories will be discussed briefly in relation to the issues in designing them, their speaking environment and human factors.

- Command and control

Command-and-control applications were originally developed for the manipulation of machinery, but have expanded to several other applications such as architecture, automotive, consumer products, manufacturing, healthcare, military, mining and telecommunications, among others. It is used for support systems where the hands are preoccupied with equipment. The alternative use of voice on the support tasks (while concentrating on the main task) enhances productivity, safety and accuracy. Pilots, for example, can command a visual display using voice while keeping their hands on the other aircraft controls. Speech also enables disabled people to rely on themselves and to depend less on their colleagues in carrying out their tasks.

The vocabulary required and the structural complexity of command-and-control systems depends on the equipment to be controlled. Usually, the vocabulary can be anything from 20 words up, depending on the number of operations that can be performed in the application. For a typical Video Cassette Recorder (VCR), the vocabulary can be about 70 words and for a military aircraft it can be 1500-5000 words. Identifying words for the vocabulary that will cover all the aspects of the application can be difficult and it is better if users are enabled to control and change their respective vocabularies. Many machines are controlled manually and developers must find simple verbs to correlate with the actions performed.

If the user population consists of a small group and the system has a small vocabulary, a speaker-dependent approach will be the best speaker model to employ as it has the highest accuracy rate. But for large vocabulary systems, training will become cumbersome and that may lead to users rejecting the system. Therefore, in this case, speaker-adaptive models can be used. Telephone-based applications normally use speaker-independent models. This is the main reason why the vocabulary is usually restricted in order to have a reasonable standard of accuracy. With the concept of subword modelling, the small vocabulary constraint was relaxed. But subword modelling has compromised a lot on the accuracy of the systems because the accuracy in subword modelling system is always lower that if the same system uses full-word modelling. The commands on this kind of systems are normally bursts of keywords, therefore both discrete-word and continuous speech can be used.

Most command-and-control applications operate in noisy environments. High accuracy is nevertheless required. The interface must therefore possess error control and recovery mechanisms in case the interaction mechanism is overwhelmed by the noise. Users of command-and-control systems view speech recognition systems as part of the applications and expect immediate and accurate response to their commands. Any speech recognition system used in command-and-control applications must be simple and easy to use. Commands must be easy to remember. Disability among users must be taken into account and if necessary, more enrolment must be carried out. The speed and manner in which the users talk to the system must be known by the system (via enrolment). Speech recognition can be used in downsizing the staff. One speaker can talk to many machines simultaneously, unlike controlling with hands. But safety of users must be taken into account before such decisions are made.

- Data entry

Speech-recognition systems in data-entry applications are used to convert verbal input into text-based data for applications. Data entry was the main use of early speech technology as it offered a hands-free environment. But it has expanded to construction, dentistry, education, healthcare and insurance, to mention but a few. Speech is the only modality that works without the use of hands, eyes and significant mental application in data-entry processes. It can be combined with other modalities to compensate for the limitations of those modalities. For example, if items' names and descriptions are entered through bar codes, the condition in which that particular item is cannot be described. That is where speech interface can enable a user to provide what is missing in the use of the other modality.

Data-entry systems are also characterized by small vocabularies, mostly fewer than 50 words. The vocabularies consist mainly of names and characteristics of items to be entered. But with the ability to allow for expanding vocabularies, their full potential is about to be realized. The vocabularies consist of standardized items (digits, names, codes etc.) A greater portion of these is unique per application. Therefore, built-in vocabularies hardly provide all the words needed for a particular application. This emphasizes the need for a vocabulary development tool to be included. Input words

are usually translated into codes, e.g. the input ‘very old’ may translate to a code ‘K122’. Speech usage avoids multiple distinct entries of the same item as the vocabulary is predefined. It also does not become cumbersome and difficult as the vocabulary grows.

Most applications are structured and are therefore readily represented by finite state grammars. Structures enhance accuracy and speedy processing. However, the trade-off between accuracy and ease of use must be considered. Speaker-dependent processing, for example, is accurate but enrolment compromises its ease of use. Speaker-independent recognition, on the other hand, is easy to use and does not need enrolment but its accuracy is not as good as speaker-dependent modelling. As the vocabulary grows, enrolment becomes cumbersome and speaker dependent, however accurate, becomes almost impossible to use. In that case speaker-independent models or speaker adaptation must be used. The input to data entry is characterized by a burst of single words. The choice of the flow of speech (discrete or continuous) depends not just on accuracy but also on environmental noise and the type of users.

Most data-entry environments are noisy and harsh. They are also characterized by causes of irritation to users such as dirt, high or low temperatures, vibrations or tiredness. Many applications designed for noisy environment use speaker-dependent systems in order to attain reasonable accuracy. Speaker-dependent recognition has the advantage that it models the speakers, the noise and the Lombard effect in advance. But even then, it is important to test the application in its anticipated deployment environment.

- Information access

This function is new to speech-recognition technology and it aims at retrieving information stored on an online source. The role of the speech recognition is to guide and instruct the users on what to do and what to get from the system. The earliest information-access applications to make use of speech are banking. But the use has expanded to automotive, consumer products, customer services, equipment repair, finance, law enforcement, office systems, retail and tourism - and the list is still

growing. Most information access applications are conceptually simple (dialling a telephone number, retrieving an account balance or retrieving a telephone number). But they enabled information to be accessed inexpensively and over long distances.

The vocabularies of information-access applications are also small. They can generally be designed to consist of highly differentiated keywords. But, as with other applications, the vocabulary will also grow as the application base expands. Information retrieval can be used by users who do not want to learn, for example, Structured Query Language (SQL) commands. It can replace the complicated system terms that are used in SQL or logic with natural-language queries. Applications designed for small groups are implemented using finite state grammars. Telephone applications for one-time-users rely on carefully designed call flows, and keyword spotting. Thus, properly guided interaction ensures that the system knows exactly what kind of input to expect after every prompt.

The choice of speaker modelling depends on the nature and size of the user population. For smaller groups such as a household, speaker-dependent will be the best choice, as it can be optimised for personalized interaction (e.g. dial by name). If it is for a larger but repeated user population, speaker adaptation will be a better choice. The speech flow commonly used is continuous speech, which is combined with word spotting. Discrete-word recognition can also be used, provided that it does not cause inconvenience to the users in pausing between words.

The most common speaker environment for information retrieval is telephones. Desktop applications generally imply an office environment where the noise is unpredictable and inconsistent. Usually, background noise includes human speech, which can easily be confused with the speech input from the genuine user. Consumer and manufacturing applications have a wide range of different environments, and the designer for these applications must assume the worst-case scenario. The human factors for information retrieval resemble those of telephone applications.

- Dictation

Dictation is aimed at producing a text-based document from speech input. Document generation through speech is potentially faster than traditional typing (for non-expert typists) and hence speeds up the digitisation of information. It also helps hearing-impaired people to follow proceedings in conferences and other gatherings. The vocabularies for dictation systems often consist of the following four parts:

- The Resident Vocabulary is loaded with the application and contains all the words that are to be used with an application.
- The User Vocabulary can be a component of the resident vocabulary together with some new words that a user may add. The user vocabulary is kept in a separate file for each user
- Dictionary Vocabulary is a backup for users and is not loaded with the application. It is used when the user adds some words to the resident vocabulary.
- Active Vocabulary is the runtime list of the words that are likely to be used at that particular time.

Effective dictation requires huge vocabularies. The content of the vocabulary matters more than the size, as it is important that the content of the vocabulary conforms to the needs of the users. Addition of words to the vocabulary differs on different applications but care should be taken that the new words are spelled correctly so that spelling mistakes are not propagated into the system. The speech input is translated into written text. Many systems utilize the concept of trigger words to select an appropriate active vocabulary.

Dictation systems require flexibility in language structures. Finite state grammars and other restrictive structures cannot be used. Many systems make use of statistical language modelling. Some designers include the ability to tune the models to some language patterns after the system has been exposed to those patterns. Ideally, dictation applications are better if they use speaker-independent models but current recognition accuracy is not yet good enough to be applied to large vocabulary systems. It is because of this that the commercial speaker-independent systems work under specified conditions and on specific categories such as for male or female users, for low or high voice, etc. Most commercial dictation systems employ speaker

adaptation to tune the systems for more frequent users. They use rapid enrolment or on-the-fly adaptation, or a combination of these two approaches.

New words added by any user may remain a property of that particular user or may become part of the resident vocabulary. Many dictation systems are used in offices or laboratories where the background noise is low, but where there is significant intermittent noise. Of late, dictation has also been deployed in hospitals and other difficult environments but with limited success. The current trends suggest that the next generation dictation system will be used as embedded systems on consumer products. Users have high expectations on the systems and many a time they tend to compare dictation systems to human transcriptionists. Nevertheless, dictation is arguably the most advanced form of speech recognition technology, even though they are far human levels of performance.

If users start with excessive expectations, the systems are doomed to failure. Users who are used to rapid dictation before editing by a second person, will find the SR dictation systems challenging to use, since they will tend to expect similar results from the speech-recognition system. It will be better if a second person is allowed to edit the preliminary results of the dictation. Most users do not like discrete word input with its pauses. When the creation of a particular document requires input from more than one user, loading and unloading of user models can be time consuming. In that case it is better to use different systems and just merge the components into one document later. Any system must provide user-friendly means to validate recognition, correct errors, add new words and to navigate through the created document.

3.3 DUAL TONE MULTI FREQUENCY (DTMF)

Prior to the invention of DTMF, telephone systems used a series of clicks to dial numbers. This method is called pulse dialling. Clicks were generated by making and breaking connections on the telephone lines and could only work within the local loop (the connection between the user and the nearest telephone company's office/station). If a connection were to be set up with a non-resident user, it would involve a human intervention. There was therefore a need to invent a way of connecting long distance calls without an intervention from a human assistant.

DTMF has its roots in the Multi Frequency signalling that was developed by AT&T in the 1950s. It has been popularly known as “tone dialling” since then, and AT&T named it “touch tone’. During this period, the telephone lines were mainly used to carry voice data, and MF signalling was used to add control signals to these circuits. However, it was discovered that voice and control signals interfered with each other, thus resulting in control information being lost. And more so, the introduction of new signals would result in requirements for more distinct frequencies. Particularly, there would be a need for ten distinct frequencies for each digit (0-9), hence the invention of DTMF.

When transmitting signals in DTMF, every character pressed is transmitted as two distinct frequencies: one from the low and one from the high-frequency groups, hence its name. Table 3.1 below shows the sets of high and low frequencies as allocated to characters on a telephone keypad [Sayed].

DTMF	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Table 3.1: DTMF frequency allocation table

For example, if a digit ‘2’ is pressed, a signal containing a low frequency tone of 697Hz and a high frequency tone of 1336 Hz is transmitted. The first four alphabetical letters A, B, C and D where used to indicate the priority status of the message. “A” indicated Flash Override, B for Flash, C for Immediate and D for Priority. The A (Flash Override) has the highest priority and could override any message. The priority decreased down to D.

Looking at the frequencies in the table, it can be observed that no frequency is a sum of a difference of any other two frequencies. Furthermore, no frequency is a whole

number multiple of any other frequency. This is done specifically to ensure that harmonics do not cause digit errors.

When the signal is received at the receiver end, an analysis is done. The analysis is done either by Discrete Time Fourier Transformation (DTFT) or by Power Spectral Density (PSD) to determine which two frequency tones are present in the signal. Signals resulting from digit '2' and digit '9' pressed will result in spectra such as those in Figure 3.3 below [Sayed].

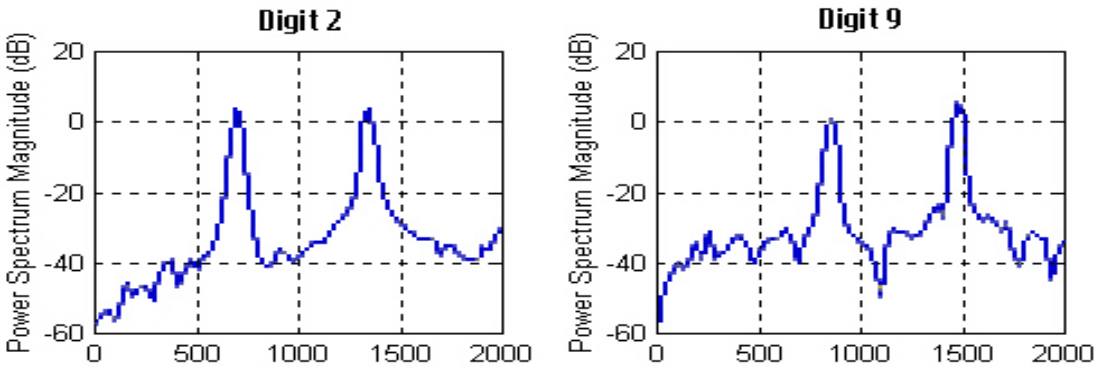

Figure 3.4: Spectral Representation of 2 and 9 respectively

The transmitter must transmit the frequencies with amplitudes within 1.5% of the design frequencies. For the signal to be accepted by the receiver, the following must apply [Giangrandi]:

- The frequencies in the received signal must not differ with the originally specified value by 3.5 %. Both frequencies must be detected correctly.
- The energies from the two frequencies must last for at least 40ms.
- Every transmission must be preceded by an energy-free interval of 40ms.
- The strength of the signal between the digits must be between –25dBm and 0dBm.
- The difference between the two frequencies must be between –8 and 4 dBm.

Otherwise, the signal will be ignored or rejected.

Having looked at the DTFM and speech recognition technology, we are ready to discuss the implementation of the two interfaces: DTFM and the speech-based interfaces.

4. DEVELOPMENT AND IMPLEMENTATION OF THE USER INTERFACES TO COMPARE SPEECH AND DTMF

The previous chapters have reviewed the fundamentals of computer interfaces, with a focus on interfaces that are suitable for telephone-based services. We now describe an experimental approach aimed at assessing these interfaces with respect to their applicability in the developing world. This work was done in collaboration with researchers from the CSIR in Pretoria, South Africa. Section 4.1 describes the experimental framework that was chosen, and section 4.2 summarizes two workshops that were held to prepare for the development of the experimental systems. Sections 4.3 to 4.5 report on the development process itself.

4.1 SCOPE OF EXPERIMENTS

The Unemployment Insurance Fund (UIF) service was chosen as the pilot project to be implemented through the CSIR's Information and Communication Technology (ICOMTEK) division. In short, the UIF process works as follows [RSA Unemployment Insurance Act 2001]: Every employer in South Africa is expected to register with the UIF at the commencement of employing any remunerated employee. The employer must furnish the UIF with all his employees' personal information and must send an updated version every month. The employer must deduct the employee's UIF contribution from the latter's salary or wages and pay it, together with the employer's contribution, directly to the UIF. As soon as the employee starts to contribute, he is entitled to apply for *unemployment, illness, maternity, adoption* and/or *dependents'* benefits, depending on the circumstance. When an employee becomes eligible, he is expected to apply for the benefits. The employer is expected to issue an Employment Service Certificate to the employee and submit it to the UIF. Upon receiving both the application from the employee and the certificate from the employer, the UIF is to process the employee's application and eventually make payments to the employee. Payments are provided monthly for a period of up to 9 months.

The aim of the project was to make these UIF services accessible to beneficiaries over the telephone. The functionalities of the current phase are limited to the

provision of the unemployment benefits. Using a standard telephone via either a touchtone or speech interface, users were able to:

- Get general information about the service
- Register for the service
- Query the status of their applications
- Check for availability of their payment and
- Re-direct their payment (cash payment or bank transfer)

In order to design and develop a usable system, the following International Standard Organization (ISO) standards were studied and referred to throughout the design and development processes:

- ISO 9241-11, which defines and describes the usability of a system and
- ISO 13407, which defines and describes the User-Centred Design Process

Two workshops were held to gather cultural and usability issues within the South African context. A paper-based prototype with relevant prompts was designed and evaluated heuristically by a panel of selected experts. The feedback from the heuristic evaluation was then incorporated into the design of the real system. The designed system was tested with real users who were identified with the help of the Department of Labour (DOL).

The UIF application is neither deep (having many steps per task) nor wide (having many options per step). That is why the call flows for both interfaces were identical, guided by an XML-based state machine.

4.2 WORKSHOPS AND THE HEURISTIC EVALUATION

4.2.1 Cultural Rendering Awareness Workshop

Understanding the cultural aspects that affect the use of software is very important in the successful deployment of many systems [Ford 03]. In order to collect views on the cultural issues to be taken into account when developing usable user interfaces in the South African context, a workshop was organized. Participants were delegates from universities, technikons, rural educational centres and national libraries. Delegates

were chosen based on their expertise in at least one of the following: computer science, sociology, development studies, languages, literacy programs, applied communications and the application of information and communication technology. The specific objectives of the workshop were to:

- Identify cultural issues to be considered in the design and implementation of a user interface in the South African context;
- Identify the cultural factors of concern in user interface design for DTMF and Speech Recognition;
- Look at other approaches that are successful and;
- Define the methodology to be followed as well as the successive steps in the development process.

Within this workshop, several key factors for the design of user interfaces in the developing world were identified. These included:

- Factors related to the particular user, such as gender, age, environmental and geographic location and functional literacy
- Factors related to the application, including accessibility, trust and buy-in by the relevant community.

Questions that should be answered if usable interfaces are to be designed and implemented within the South African context were raised. These included: which distinct groups are being targeted, what is their technological experience, and what is the current situation with respect to the type of service that is being proposed.

Emphasis was put on the differences in cultures between the urban and rural communities even within the same language groups, the functional literacy within the targeted group, the lack of information and communication technology training and facilities to the rural communities and the lack of understanding the capabilities of information and communication technology in the target audience

A survey on the Tele-centres in the Limpopo province was successfully carried out. From that experiment, the following lessons can be learned:

- Issues that facilitate/impede access to the facilities need to be understood before such innovations can be placed in communities previously unexposed to them.
- Issues that affect control on the systems need to be investigated.

The workshop emphasised the need to involve the users as well as other stakeholders in the development process. The iterative method of system development was identified as being the most suitable methodology in the development process.

4.2.2 Workshop on Human Computer Interaction

Another workshop, on human-computer interaction (HCI) was organized. Prof. Janet Wesson of the University of Port Elizabeth, who is one of the foremost HCI and usability experts in South Africa, guided the discussion. The aim of the workshop was to:

- Develop a better understanding for the underlying human computer interaction issues on user interface design in general;
- Develop a better understanding of the human computer interaction design issues specifically for DTMF and Speech Recognition interfaces;
- Learn about successful approaches and initiatives related to our system, and;
- Derive a minimum set of requirements needed for the design of these interfaces.

From this workshop, the importance of system usability to address the discovered cultural constraints was revealed. The workshop identified usability goals, user experience goals and user pragmatic goals that are important to the design and successful implementation of telephony interfaces. Deliberations were centred on effectiveness, efficiency and satisfaction, and how they can be measured on telephony interfaces when being used in the South African context.

The aim must be to develop a system that users can use to achieve their goals with minimal or no help. They must also feel that it is easier to use the system than to use the already existing methods.

The University of Port Elizabeth carried out a local research in which the created user profiles of students. They found that:

- Culture is a complex concept and it is therefore difficult to quantify.
- The best practice is to map goal and tasks to something that users do in real life.
- Experience with any kind of technology enhances the users' ability to use other systems.

4.2.3 Heuristic Evaluation

After gaining an improved understanding of relevant cultural and usability issues, a paper-based prototype system was designed. This consisted of a set of call-flows and the associated prompts ("scripts"). A panel of ten experts was chosen to evaluate the system heuristically. The usability problems identified by these experts concentrated mainly on error messages, error prevention techniques and naturalness of the call flow and, to a lesser extent, the content of the scripts with respect to length and politeness. All these issues were considered during the second iteration of the development cycle.

4.3 STRUCTURE OF THE SYSTEM/INTERFACES

With input from the heuristic evaluation, the pilot experimental system was designed. Below is a brief description of the structures and characteristics of the two components, one accessed via key-presses ("DTMF") and the other via speech recognition.

4.3.1 Structure of the DTMF System/Interfaces

To access the system, the user must dial a given telephone number to get connected to the system. Upon receiving the call, the system plays the greeting prompt, gives brief information and instructions on the subsequent steps and waits for user input. The user enters the DTMF commands by pressing the relevant keys on a standard telephone. These commands are transmitted via the telephone network to the system, which in turn interprets them. The system changes its state accordingly, following an XML-structured flow of events. It plays pre-recorded prompts back to the user, based on the current state and again waits for the user input. The cycle is repeated until the user exits the application. Figure 4.1 below shows the structure of the DTMF based system.

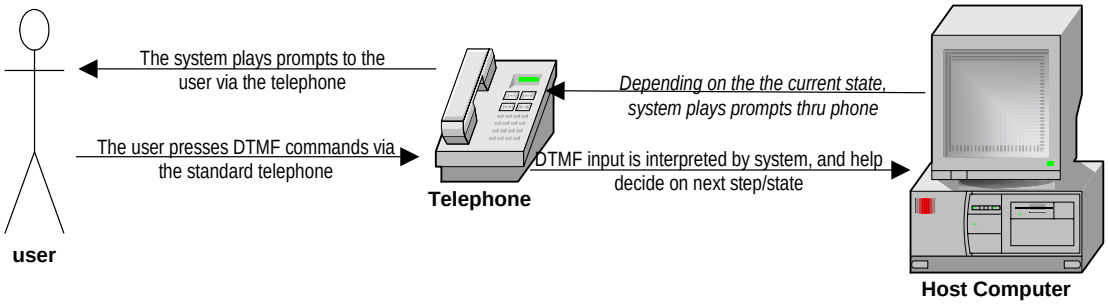

Figure 4.1: The DTMF system Structure

4.3.2 Structure of the Speech Recognition System/Interface

Since it was not feasible to design speech-recognition systems for the languages of interest, a wizard-of-oz structure was employed (where a person performs the work of a speech recognition system, i.e. interprets the speech input and enters the relevant commands onto the system). As with the DTMF component, the user dials a number to get connected to the system. The system plays back the preliminary prompt and waits for the user input. The user enters the speech input via the standard telephone. The 'wizard', an operator who acts as an intermediate between the user and the application, listens to the speech input. Using a graphical user interface (GUI), the wizard then enters the corresponding input onto the application. Upon receiving the relevant input, the application changes itself from one state into another, depending on the conditions, following a pre-defined XML-based process flow. The application

itself was situated some kilometres away from the users, thus imitating the real life transmission delays of the telephone channels. Figure 4.2 below illustrates the set up.

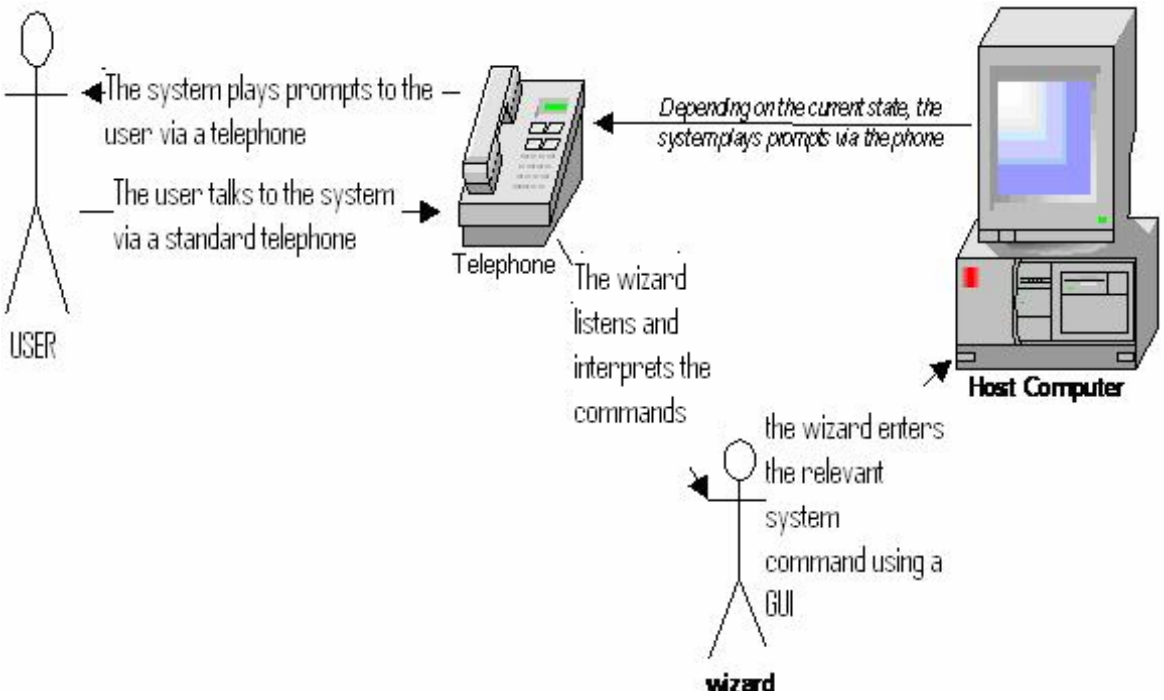

Figure 4.2: The Wizard of Oz Structure

4.4 CALL FLOW, PROMPTS AND REQUIRED USER INPUT

In recording the prompts, a female voice was used, to ensure similarity with the current help line of the UIF offices. The prompts were designed for maximum continuity, each prompt thus picking up where the previous one has left off. Prompts were kept short, unambiguous and provided in a clear voice. At any point, the DTMF choices were limited to a maximum of four. The industry standards and conventions were adhered to as much as possible. For example, like in any other call based conversation, if the user wants to terminate the conversation, he would just hang up the phone.

The user provided input to the system via a standard telephone, either by pressing keys or by speaking. The speech input was limited to YES/NO answers, except in a few cases e.g. entering of an ID number or where there are more than two choices. Easy-to-understand terms for choices were used wherever possible, and the user was informed, at the beginning of the session, what to do should he need to speak to

a human assistant. Addendum A gives the list of prompts for the touch tone interface while Addendum B gives the prompts for the speech based interface.

4.5 PLANNING AND EXECUTION OF EVALUATION

The system evaluation was carried out at one of the labour centres in Pretoria. Participants were chosen from the actual beneficiaries who were queuing for the UIF service. The computer hosting the application was kept at the CSIR offices, some 15km away from the testing venue.

4.5.1 The Participants

One important aspect in the successful implementation of a system for customer service is to know the users and the tasks they are to carry out using the system. According to Kotelly [Kotelly 02 pp. 127], testing is only meaningful if the participants represent a fair representation of the actual users who will use the system. This particular application is meant for adults who may have little or no computer exposure, but who are able to use a standard telephone. Most of the intended users have lost their jobs and are therefore likely to be under stress. Such users do not have very much patience; hence, they need to achieve their goals as quickly as possible. It has to be noted that the system just facilitates the process of applying for money and does not actually give out money. It is not expected that the users are familiar with the system, and hence they require easy-to-follow instructions throughout the process.

The interface was tested in two different languages, Setswana and English. A balanced representation with respect to language and gender was strived for in choosing the participants. Other factors such as age, level of education and experience in using similar systems were also considered but could not be used as a basis for the selection of participants. These factors could only be determined after the participants were selected, that is, during the interview.

4.5.2. The Tasks

The participants were asked to carry out any of the following tasks using the system:

- Register for the service
- Check availability of payment and
- Change their payment details (redirect their payments to cheque or bank transfer)

All these tasks have approximately the same number of steps as well as choices under each step. They are therefore expected to take approximately the same amount of time.

4.5.2 The Process

The process of collecting data consisted of three parts: The pre-test interview, the actual test of the system and the post-test interview. Since some of the participants were not sufficiently literate to complete questionnaires themselves, an interview was conducted individually for all the participants. The pre-test interview was aimed at collecting statistical data on the participants, e.g. the age, gender, etc. Bearing in mind that there were two different languages, the interview was conducted in the language that the participant chose to operate in. The participant was informed about the aim of the whole process at the beginning of the pre-test interview.

After the pre-test interview, the interviewer briefed the participant on the functionalities of the system and what tasks they were expected to fulfil. They were then allowed to carry out a task using the system. The participants were observed by members of the experimental team. Unless otherwise requested by the participant, no assistance of any kind was rendered.

After using the system, the participant was taken to a different room for a post-test interview. The aim of the post-test interview was to obtain feedback on the user's experiences in using the system.

5. RESULTS AND DISCUSSION

5.1 THE PARTICIPANTS

A total of 31 participants tested the two interfaces: 17 tested the DTMF and 14 tested the speech interface. For the sake of clarity, participants will be referred to as 'participant 1', 'participant 2' etc. depending on the chronological order in which they used the system.

Out of the 17 DTMF interface participants, 9 were males and 8 females. Their ages range form 22 to 54 with an average age of 34. 12 were Africans, 3 Caucasian and 2 Indians. 6 participants could not speak English. All of them could use at least one electronic device (telephone, video etc). 10 have lower than Grade12 qualifications, 2 have Grade 12 and 5 have tertiary qualifications. 6 participants were familiar with the UIF process while 11 were not.

Out of the 14 speech interface participants, 8 were males and 6 females. Their ages range from 21 to 55 with an average age of 37. One participant's age was undetermined. 10 were Africans and 4 Caucasian. 1 participant could not speak English. As with DTMF, all of them had been exposed to at least one electronic device. 7 have lower than Grade 12 qualifications, 3 have Grade 12 and 4 have tertiary qualifications. 8 participants were familiar with the UIF process while 6 were not.

5.2 PERFORMANCE

5.2.1 Completion rate

Of the 17 participants who used the DTMF interface, 10 (59%) completed their intended tasks. The 7 failed processes can be described as follows:

(i) 3 participants kept requesting the same information repeatedly, and then hung up;

(ii) 1 participant (when requested) kept on entering an invalid reference number before hanging up;

(iii) 1 participant replaced the phone handset because the prompts were not clear to her;

(iv) 1 participant replaced the handset when asked to enter his ID number, because of a security concern regarding his ID number;

(v) 1 participant requested all four options which provide information on the system itself, then pressed " * " for human operator and immediately replaced the handset.

Of the 14 participants who used the speech interface, 9 completed their intended tasks, which is equivalent to 64%. The 5 failures can be described as follows:

(i) For the first 3 participants, the system simply went down in the middle of the call;

(ii) 1 participant ended the call as she was not literate enough to enter her ID number;

(iii) 1 participant dropped the phone after successfully entering her ID number. She explained that she had thought her task was just to provide that number. (That is, the goal of the experiment had not been clear to her).

Note that the first 4 failures have little to do with the interface: 3 are pure application system errors and the fourth user would not be able to enter the required information regardless of the interface (in fact, she was not able to provide the ID number to the human telephone operator). Therefore, a task completion rate of 8/9 = 89% could justifiably be claimed for this interface. Similarly adjusting the DTMF success rate to count only interface failures, gives a task completion rate of 10/15 = 67%. Thus, the speech interface seems more successful from a task-completion perspective.

5.2.2 Completion Time

The time taken by the participants using the DTMF interface (who completed their tasks) ranges from one to seven minutes inclusive, with an average of three minutes. The speech interfaces participants took times between three and five minutes inclusive, with an average of four minutes. Three of the DTMF interface participants recorded shorter periods than any of the speech interface periods (one, two and two

minutes respectively), while two others recorded the two longest periods (six and seven minutes respectively). The others took exactly three minutes.

Of the shortest times, all three concerned participants were educated to at least Grade 12. All had experience with a wide variety of electronic equipment, have cellular telephones and are experience in using both their cellular telephones and Automatic Teller Machines (ATMs). It can thus be assumed that these participants already know how to use a touchtone-based interface. In addition, unlike the rest of the participants, these participants did not ask for any information on how the system works. They went directly to the tasks they wanted to fulfil, completed those, and disconnected immediately. Therefore, they could not be categorized as being technologically unsophisticated.

The two participants who took the longest time to complete their tasks both own and know how to operate a radio, a TV and a cell phone. However, their formal education is less than grade 12 and both have no experience with the UIF process. Interestingly, one is the youngest and the other is one of the two oldest participants.

The completion periods for the speech interface participants are evenly distributed, two participants took three minutes, five participants took four minutes and the other two took five minutes.

Although the average time taken to complete the tasks is shorter on the DTMF (3 minutes), it must be noted that this interface has also recorded the two longest times. Furthermore, the average is highly influenced by the three people who could not be categorized as technologically unsophisticated. Therefore, we are not able to draw a firm conclusion from this data.

5.2.3 Error analysis

To gain a better understanding of the errors that occur when using the two systems, we have analysed the errors made by participants while providing their ID numbers to the system.

Of the 17 participants who used DTMF entry, four ID entry errors were recorded (24%). These four participants all possess a qualification of less than Grade 12. The errors can be described as follows:

- A participant omitted two successive characters;
- Two participants omitted one character;
- A participant entered two incorrect characters;

On the speech interface, two errors were recorded, which is equivalent to 14%. The two concerned participants have tertiary and Grade 12 qualifications respectively. All errors are cases where a user entered a totally wrong ID and can be described as follows:

- Participant 3 entered a totally wrong ID, (entered a three-digit "111", instead of the thirteen digit ID).
- Participant 3 entered a wrong eighteen-digit combination of 0s, 1s and 2s

It therefore seems as if the errors that occurred when using the speech system were of a different nature: specifically, they involved mistaken choices of numbers, rather than incorrect entry of a known number.

5.2.4 User Satisfaction

Participants were not given the opportunity to compare the two interfaces (although such a comparison is planned for future experiments, it was not feasible for users to interact with both systems during our initial experiment). Nevertheless, they were given a chance to express their satisfaction on the interface they tested.

On the DTMF interface:

(i) 1 participant felt there is a need to give feedback/confirmation on what keys were pressed;

(ii) 1 participant felt that at some points she did not know what to do;

(iii) 2 participants expressed security concern on the system (including one who refused to provide his ID);

(iv) 2 participants felt that the system must have a mechanism to repeat the information it provides to the user;

(v) 1 participant felt that the prompts were too long and;

(vi) 1 participant felt there should be some form of volume control on the interface.

On the speech interface:

(i) 1 participant felt that at times she could not understand the prompts/questions;

(ii) 1 participant felt that it is too much to ask the users to be at a quiet place, more especially if they are using public phones;

(iii) 1 participant felt that the pauses between prompts are too long and preliminary music should be provided on the system.

In total 5 different complains were made by the 17 participants using the DTMF interface.

Our subjective assessment was that users of the speech interface looked more comfortable and some did not even realise that they were dealing with pre-recorded voices. They assumed they were interacting with a human operator on the other end. In general, the speech interface participants were more satisfied.

5.3 OTHER ISSUES

The ideal case was to have an equal number of participants, and to have the same participants testing both interfaces. However, none of the participants tested both interfaces and fewer participants were obtained for the speech interface (14 as opposed to 17 on the DTMF interface). (On the testing date of the speech interface, the staff at the Department of Labour stopped processing applications for the beneficiaries earlier than expected. As a result, the queue from which participants were canvassed dispersed.)

Another unexpected problem at the test site was that the voice prompts were not sufficiently loud (they had previously been tested from internal telephones at the development site, and the experimental team had not realized that there would be

such a loss of amplitude externally). It was therefore impossible to use the speakerphone for observation purposes, as had been planned. Also, almost all the participants complained about the volume of the prompts.

5.4 SUBSEQUENT EXPERIMENTAL FINDINGS

After collecting and analysing the data that was collected at the Labour Centre in Pretoria, the usability shortcomings that were identified on the interfaces were rectified. A subsequent experiment was carried out in Stanger, Kwazulu-Natal. By then, a translation in Isizulu was implemented. There were thus two versions of the same interfaces, one in English and the other one in Isizulu. Though the results obtained from this experiment are not the focus of this research, some findings from this experiment are worth mentioning here.

Kwazulu-Natal is mainly populated by the Isizulu speaking people. Many participants who used the English version did so on the request of the experiment team. Otherwise they preferred to do the interaction in their home languages, Isizulu. It can therefore be deduced that users prefer to interact in their home languages.

Some participants pointed out that if the application requires the beneficiaries to sign papers, then the telephony systems do not really solve the problem because in any case the users will have to visit the host offices. Care should therefore be taken when choosing the interface for a particular application.

Some participants who struggled to successfully provide their ID to the application using the speech interface reverted to using the touchtone one, i.e. entering the ID using the telephone keys. There was also a problem with the format in which the numbers should be read. For example, the number '*12*' is read as '*one two*' by some participants while others read it as '*twelve*'. The loudness of prompts improved considerably even though there were still a few participants who struggled to hear and understand them.

6. CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The purpose of this research was to compare the two interfaces, DTMF and speech-based, with respect to transaction completion rate, time taken to complete transactions, error rate and user satisfaction. In all but 'time taken to complete transactions', there is evidence that the speech-based interface has out-performed the DTMF interface. This conforms with the widespread belief that speech interfaces will serve as a bridging gap in enhancing accessibility of information to technologically unsophisticated users. It can therefore be concluded that when the user population consists of mainly technologically unsophisticated users, the speech interface is more effective and more satisfying than the touchtone one.

Neither of the interfaces could address the issue of innumeracy on the side of the users. Therefore, unless entry of numeric data (ID) is removed from the application, this issue remains a problem to both interfaces.

6.2 SUGGESTED FUTURE RESEARCH

As a result of this research, further exploration in the topic has been suggested. This includes some suggested improvements that can be made to this research, i.e. collecting and comparing data on the same parameters, but under different conditions or using different sets of users. The second suggestion is the set of other parameters that can be measured and compared on the same interfaces.

The ideal case would be if the same users were given a chance to test both interfaces. Care should be taken in order to avoid one interface being tested first by all users. The second suggested improvement is to use different languages i.e. having the systems in duplicate languages in order to see whether languages have any effect on the performance of the interfaces. The third suggested improvement is to employ users who have more or less the same technological exposure but with different levels of education. This will be done in order to see whether the level of

education has any effect on the performance of these interfaces. The fourth and the last suggested improvement is to make an objective evaluation of the user satisfaction.

Data entry rate is the suggested additional parameter that can be measured. This will answer the question of which of the two interfaces will enable the users to enter data (e.g. the ID) at a faster rate.

REFERENCES:

1. Anderson W. H. (2002), *Speech Recognition 2002, Real Uses in Real Life,* Aba Techshow 2002, http://www.wellslegaltech.com/SpeechRecognition2002.htm (Latest access on 12 May 2004)
2. Bigler J (2003), *The Dvorak Keyboard*, http://www.mit.edu:8001/people/jcb/Dvorak/index.html (Latest access on 10 July 2004)
3. Brain M, (2002), How Microprocessors Work, http://computer.howstuffworks.com/microprocessor.htm/printable (Latest access on 17 May 2004)
4. Brain M, (2003), How Computer Monitors Work, http://computer.howstuffworks.com/monitor.htm (Latest access on 12 July 2004)
5. Brain M, (2003), *How Computer Mouse Work,* http://computer.howstuffworks.com/mouse.htm (Latest access on 10 July 2004)
6. Dix, Finlay, Abowd & Beale (1998), *Human-Computer Interaction*, 2nd edition, Prentice Hall
7. Drissman A (1997)*, Handwriting Recognition Systems: Overview*, http://www.drissman.com/avi/school/HandwritingRecognition.pdf (Latest access on 13 July 2004)
8. Engle J (2001), *Speech Recognition Primer*, Netbytel Inc. http://www.netbytel.com/e-gram/egramdetail.asp?ID=109 (Latest access on 11 June 2004)
9. Ford G, Gelderblom H, *The effect of culture on performance achieved through the use of HCI*, Proceedings of SAICSIT 2003, pp 218-230
10. Giangrandi I, *The DTMF Encoding*, http://www.giangrandi.ch/jack/radio/dtmf-e.shtml (Latest access on 12 January 2004)
11. Gilman D.J., *Trackballs*, http://www.abilityhub.com/mouse/trackball.htm, Latest access on 11 July 2004)
12. Hewett T.T. et al (1997), *Curricula for Human Computer Interaction*, http://sigchi.org/cdg/cdg2.html#2_1 (Latest access on 12 June 2004)
13. Jedruszek J, *Speech Recognition*, Alcatel Telecommunication Review, 2nd Quarter 2000, http://www.alcatel.ru/news/publications/files/speech_recognition.pdf (Latest access on 4 July 2004)

14. Karat J. (2003), *The Evolution of Human Computer Interaction*, Proceeding of the CHI-SA 2003, http://www.chi-sa.org.za/CHISA2003 (Latest access on 16 April 2004)
15. Konig Y and Morgan N (1993), *Supervised and Unsupervised Clustering of the Speaker Space for Connectionist Speech Recognition*, Proceedings of the International Conference on Acoustic, Speech and Signal Processing 1, p 545-548
16. Kotelly B, *The Art and Business of Speech Recognition*, Addison-Wesley Pub Company, 2002
17. Liebowitz S. et al, *Typing Errors*, Reasononline, http://reason.com/9606/Fe.QWERTY.shtml, (Latest access on 10 May 2004)
18. Markowitz J. (1996), *Using Speech Recognition*, Print Hall PTR
19. Picone J, Price P (2000), *Can Advances In Speech Recognition Make Spoken Language As Convenient And As Accessible As Online Text?* http://www.isip.msstate.edu/publications/conferences/aaas/2000/speech_recognition/ppt/patti.ppt (Latest access on 13 July 2004)
20. Preece J, Rogers Y, Sharp H, Benyon D, Holland S & Carey T (1994), *Human-Computer Interaction,* Addison Wesley Publishing Company
21. *Republic of South Africa Unemployment Insurance Act-2001*, http://196.25.215.100/legislation_detail.asp?LegislationID=3 (Latest access on 14 July 2004)
22. Sayed A H, *Touchtone Telephone Overview*, UCLA Electrical Engineering Department, http://www.ee.ucla.edu/~dsplab/ttt/over.html, (Latest access on 8 July 2004)
23. Strandberg T et al (2003), *The Microprocessor*, http://www.raptureready.com/rap31d.html (Latest access on 17 May 2004)
24. Tyson J (2003), How RAM works, http://computer.howstuffworks.com/ram.htm (Latest access 14 July 2004)
25. Tyson J (2003), How ROM works, http://computer.howstuffworks.com/rom.htm, (Latest access on 5 July 2004)
26. Tyson J (2003). How LCDs work, http://electronics.howstuffworks.com/lcd.htm
27. Wikipedia, *Dual Tone Multi Frequency*, http://www.wikipedia.org/wiki/Touch_tone, (Latest access on 14 July 2004)
28. Winograd T (1983), *Language as a cognitive process*, Addison-Wesley Publishing Company, 1983

ADDENDUM A

Scripts for a DTMF-based interface to the UIF grant system

Number	Script
1	Welcome to the Unemployment Insurance Fund grant service. Thank you for using this service.
2	If you need more information on the Unemployment Insurance Grant process please press 1. If you want to continue please press 2. If at any point you want to end this service, please hang up.
3	You are about to receive further information on the UIF grant service. • To find out who is eligible for a UIF grant, press 1. • To find out how to apply for a UIF grant, press 2. • To get more information on this service, press 3. • To continue with the UIF process please press 4
4	The following individuals are eligible for UIF grant payments: • All unemployed contributors who have not voluntarily resigned. • Any worker who has been laid off because of ill health • Any dependents of deceased contributors • Parents adopting children below two years of age • Pregnant women who are now able to claim maternity benefits without jeopardizing their unemployment benefits.
5	• You can apply for your UIF grant using this service. • In addition, you need to visit your nearest Labour centre for verification, in the interests of security. Please take with you, your ID document and your last 6 payslips to date. • Using your allocated reference number, provided by the service, you can check the progress of your grant application and payment details, using this service.
6	To use this service successfully, follow the prompts given by the speaker. If at any time you wish to go back to the previous menu, press *. To speak to an operator at any time, press 0. To end this service, please hang up.
7	Please enter your 13-digit ID number
8	The number you have entered is not a valid ID number.
9	The ID number you have entered is: (*ID number*)
10	If this is correct, press 1. If this is not correct, press 2.
11	Your ID number has successfully been captured
12	If you don't have a reference number please press 1. If you have already applied for a UIF grant and you have received a reference number please press 2.
13	Please enter your UIF application reference number, followed by a #
14	The reference number you have entered is
15	If this reference number is correct please press 1

	If this reference number is incorrect please press 2
16	The reference number you have entered is invalid
17	If you want to re-enter your reference number please press 1. If you want to apply for a UIF grant please press 2
18	The reference number you have entered is correct
19	If you want to track your application status please press 1 If you want to check your grant availability please press 2
20	Your UIF application number is
21	If you want to repeat the reference number, please press 1. To continue please press two
22	Thank you for applying for your UIF grant. We have not yet received your Employment termination notification, but your application is successfully captured. For future enquiries please use the UIF application reference number provided.
23	Thank you for applying for your UIF grant. We have received your Employment termination notification, and your application is successfully captured. For future enquiries please use the UIF application reference number provided.
24	This service allows you to track the status of your UIF application
25	
26	Your employment notification has not yet been received.
27	Your employment termination notification has been received and is in progress
28	Your UIF application has been approved
29	If you want to know your next payment date please press 1. If you want to exit this service please press 2
30	Thank you for using the UIF grants application and tracking system. Please remember to keep your reference number safe, as you will need it to use this service again
31	There seems to be a problem with your UIF application. We will contact you in due course with regard to your application.
32	This service will inform you of your next UIF grant payment date
33	Your next UIF grant payment will be ready on the following date:
34	Unfortunately the next payment date is not available yet. Please contact us again within the next week to check whether the payment date is available.
35	You can collect your grant payout from your nearest labour office
36	If you would like us to transfer the money directly to your bank account please press 1. If you want to collect the grant payout from you nearest labour office please press 2.
37	Your money will be transferred to you on the:
38	You are being transferred to the operator

ADDENDUM B

Scripts for a Speech-based interface to the UIF grant system

Number	Script
1	Welcome to the Unemployment Insurance Fund grant service. Thank you for using this service. In order to use this service effectively, please make sure you are in a quiet place with no background noise. Please answer my questions clearly and directly and only speak when you are ready to answer. If at any point you want to speak to an operator, please say ”OPERATOR”, and to return to the main options, please say “MAIN OPTIONS”
2	If you need information on the Unemployment Insurance Grant process please say “INFORMATION”. If you want to continue please say “CONTINUE”. If at any point you want to end this service, please hang up.
3	You are about to receive further information on the UIF grant service. • To find out who is eligible for a UIF grant, please say “ELIGIBLE”. • To find out how to apply for a UIF grant, please say “APPLY”. • To get more information on this service, please say “MORE INFORMATION”. • To continue with the UIF process please say “CONTINUE”
4	The following individuals are eligible for UIF grant payments: • All unemployed contributors who have not voluntarily resigned. • Any worker who has been laid off because of ill health • Any dependents of deceased contributors • Parents adopting children below two years of age • Pregnant women who are now able to claim maternity benefits without jeopardizing their unemployment benefits.
5	• You can apply for your UIF grant using this service. • In addition, you need to visit your nearest Labour centre for verification, in the interests of security. Please take with you, your ID document and your last 6 payslips to date. • Using your allocated reference number, provided by the service, you can check the progress of your grant application and payment details, using this service.
6	To use this service successfully, follow the prompts given by the speaker. If at any time you wish to go back to the previous menu, say “MAIN OPTIONS”. To speak to an operator at any time, say “OPERATOR”. To end this service, please hang up.
7	Please state your 13-digit ID number. When finished, please say “CONTINUE”
8	The number you have provided is not a valid ID number.
9	The ID number you have provided is:
10	If your ID# is correct, please say YES

	If your ID# is not correct please say NO
11	Your ID number has successfully been captured
12	If you don't have a reference number please say "NO REFERENCE #". If you have already applied for a UIF grant and you have received a reference number please say "CONTINUE".
13	Please state your UIF application reference number. When finished, please say "CONTINUE"
14	The reference number you have provided is
15	If this reference number is correct please say YES If this reference number is not correct, please say NO
16	The reference number you have provided is invalid
17	If you would like to re-enter your reference number please say, YES If you do not want to re-enter your reference number please say NO
18	The reference number you have provided is correct
19	If you would like to track your application please say TRACK APPLICATION If you would like to check grant availability please say AVAILABILITY?
20	Your UIF application number is.. Please keep it safe for future inquiries.
21	If you would you like me to repeat the reference number please say YES If you do not want me to repeat the reference number please say NO
22	Thank you for applying for your UIF grant. We have not yet received your Employment termination notification, but your application is successfully captured. For future enquiries please use the UIF application reference number provided.
23	Thank you for applying for your UIF grant. We have received your Employment termination notification, and your application is successfully captured. For future enquiries please use the UIF application reference number provided.
24	This service allows you to track the status of your UIF application
25	Your employment notification has not yet been received.
26	Your employment termination notification has been received and is in progress
27	Your UIF application has been approved
28	If you would like to check your next payment date please say, YES If you do not want to check your next payment date please say NO?
29	Thank you for using the UIF grants application and tracking system. Please remember to keep your reference number safe, as you will need it to use this service again
30	There seems to be a problem with your UIF application. We will contact you in due course with regard to your application.
31	This service will inform you of your next UIF grant payment date
32	Your next UIF grant payment will be ready on the following date:
33	Unfortunately the next payment date is not available yet. Please contact us again within the next week to check whether the payment date is available.
34	You can collect your grant payout from your nearest labour office
35	If you would like us to transfer the money directly to your bank account please say "TRANSFER".

	If you want to collect the grant payment from you nearest labour office please say “COLLECT”.
36	Your money will be transferred to you on the:
37	You are being transferred to the operator