Design of an Advanced Text-To-Speech System for Afrikaans

Prepared by: Francois Rousseau

Supervised by: Associate Professor Daniel Mashao

Submitted to the faculty of Engineering and the Built Environment, University of Cape Town, in fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

October 2005

Declaration

I declare that this thesis is my own work, and that it has not been submitted for a degree at any other university. The work was executed by me except where acknowledgements are made referring to work done by others.

_____ October 2005

Francois Rousseau

Acknowledgements

First and foremost I would like to thank our Heavenly Father for giving me the strength and courage to achieve my greatest goals. Without you nothing is possible. To my supervisor Professor Daniel Mashao, I could not have asked for better supervision. Not only have you guided me through this thesis, but you have also taught me very valuable lessons in life that I shall take throughout my career and for that Sir, I thank you. To my family and friends, especially my mother I would like to say that I love you and thank you for all the support that you have given me through out my schooling career.

To all the members of the STAR, Speech Technology And Research group I would like to say thank you "fifty nine thousand million" times for making these two years the best years of my life. Each one of you holds a very special place in my heart. To the members of CRG outside STAR I would also like to thank for their friendship and support. To my two best friends, "Zee" and "Ree", words can not describe our friendship and it means the world to me. Finally I would like to thank my sponsors TELKOM PTY (LTD) for their financial support.

Abstract

Afrikaans is the home language to approximately six million people in South Africa. The need for an Afrikaans TTS system comes with the growing interest in integrating speech technology in all eleven official languages of the country. The ultimate goal here is to enable communication between man and machine using speech. This can be achieved with the use of speech technology by implementing multilingual technological systems that all the people of South Africa can understand and relate to.

Understandability, flexibility, naturalness and pleasantness are the requirements of an advanced TTS system. The technique of concatenative speech synthesis has been the most successful technique in meeting all these requirements. The Festival speech synthesis system uses two popular concatenative synthesis techniques to design new TTS systems in different languages. The techniques are: diphone concatenative synthesis (DCS) and unit selection synthesis (USS). Diphone concatenative synthesis has the ability to produce TTS systems with a high degree of flexibility, but often lacks in understandability, naturalness, pleasantness. Limited domain unit selection synthesis (Ldom USS) is a unit selection technique that can produce TTS systems with a high degree of understandability, naturalness, and pleasantness. The technique can however not produce the flexibility that is required since it works from limited vocabularies. Open domain unit selection synthesis (Odom USS) is the second unit selection technique, and has the ability to produce TTS systems with the same degree of flexibility as DCS technique but with a higher degree of understandability, naturalness and pleasantness.

Each of the three techniques was implemented in the aim of designing the advanced TTS system for Afrikaans. Each system was tested using subjective listening tests to show how well each system measures up to the requirements of an advanced TTS system. Results show the Ldom system outperforms the DCS and Odom systems in terms of understandability, naturalness and pleasantness. The Odom system performed acceptably

in terms of these requirements, but was inconsistent at times and the DCS system performed unacceptably. The advanced TTS system was achieved by using a hybrid approach to TTS synthesis. This approach adds flexibility to the Ldom system by using a suitable back up voice to synthesize words that are not in the vocabulary of the limited domain. Both the DCS and the Odom systems were implemented to accomplish this task. The hybrid limited-open domain system performed best since the voice quality of the open domain system is much greater than that of the diphone system. Therefore, the hybrid Ldom/Odom Afrikaans TTS system is implemented as the advanced Text-To-Speech system for Afrikaans.

Key words: concatenative speech synthesis, TTS, understandability, flexibility, naturalness and pleasantness, hybrid TTS

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

TTS	Text-To-Speech
CSS	Concatenative Speech Synthesis
DCS	Diphone Concatenative Synthesis
USS	Unit Selection Synthesis
Ldom	Limited Domain (pronounced L-dome)
Odom	Open domain (pronounced Oh-dome)
SAMPA	Speech Assessment Methods Phonetic Alphabet
VODER	Voice Operating Demonstrator
VOCODER	Voice Coder
DAVO	Dynamic Analog of Vocal tract
PAT	Parametric Artificial Talker
CHATR	Collective Hacks from the Advanced Telecommunications Research Laboratories
PDA	Personal Digital Assistant
SMS	Short Message Service
CSTR	Centre for Speech Technology Research
MLDS	Multi-Level Data Structures
GCC	GNU Compiler Collection
LTS	Letter To Sound
CART	Classification And Regression Trees
F0	Fundamental Frequency
DTW	Dynamic Time Warping
MFCC	Mel Frequency Cepstral Coefficients
HMM	Hidden Markov Model

CHAPTER 1

Introduction

Speech synthesis or text-to-speech (TTS) as it is most commonly known is becoming part of our daily lives. The ability of these systems to convert strings of organized text into a spoken output makes these systems very valuable in terms of applications for the real world. The applications range from telecommunications to hand held digital devices to assist the visual and hearing impaired, language education and electronic services to name a few. The need for a TTS system in Afrikaans comes with the growing interest in integrating speech technology into modern South Africa. The main aim is to improve productivity of the citizens and improve their lives by increasing their comfort with using modern technology. This is done by using speech technology to implement multilingual technological systems that users can understand and relate to. From a speech synthesis point of view the goal of this thesis is to investigate different techniques in designing TTS systems, and to provide a technique for the design of an advanced TTS system in Afrikaans. An advanced TTS system must be flexible, understandable, natural and pleasant to listen to.

1.1 Overview of Speech Synthesis

Verbal speech, with the exception of the hearing impaired, is the dominant form of communication between humans. We use it to communicate general information such as ideas, plans, feelings, point of views and just about any information that we can share audibly. Speech technology has extended the values of human speech communication by allowing us to communicate with computers and computerized machines in the same manner that we would communicate with other humans. Speech synthesis, speech recognition and

speech coding are the areas that make up Speech Technology [1]. This is illustrated in Figure 1.1 which also emphasizes the area of research that is investigated.

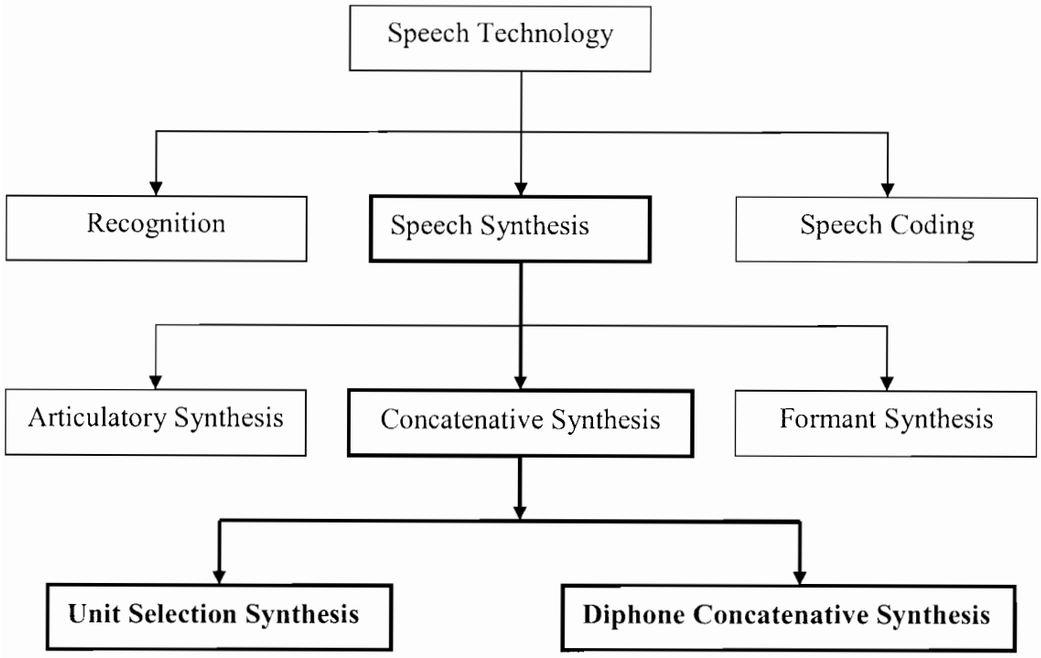


Figure 1.1: Overview of speech technology with emphasis on Concatenative Speech Synthesis [1]

Speech synthesis is task of producing speech in an artificial manner [2]. Several techniques to accomplish this task exist of which each technique falls into one of the following categories: articulatory speech synthesis, formant speech synthesis and concatenative speech synthesis. Articulatory speech synthesis attempts to model the human speech production system, formant speech synthesis attempts to model the formant frequencies of natural speech and concatenative speech synthesis (CSS) concatenates pre-recorded units of natural speech for synthesis [3]. Each of these techniques is discussed in detail in Chapter 2.

Speech synthesis has been an interest of mankind for centuries. The first attempt was completely mechanical. The system was designed by a Russian Professor called Christian Kratzenstein in Petersburg in 1779 [3] [4]. Even though the system could only produce five long vowels sounds it was this breakthrough in technology that lead to the speech synthesis

systems that we are currently using today. Computers have opened the doors to materializing the ideas of building advanced TTS systems that are fully understandable, flexible, natural and pleasant [5]. This is discussed in more detail later in this chapter.

The modern speech synthesis technique is simply the process of converting any input text into a speech output using a computerized system [6]. This process consists of two phases. Firstly, a high level phase called the front-end, and secondly a low level phase called the back-end [7]. The synthesis process and its two phases are shown in Figure 1.2 below.

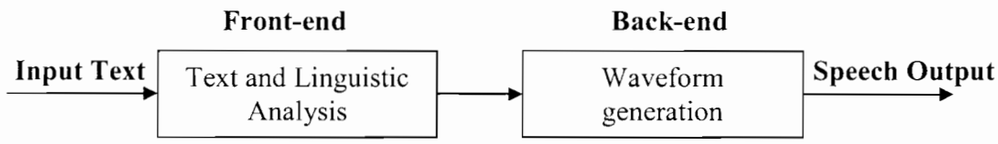


Figure 1.2: High level and low level phases of the synthesis process [6]

The front-end is responsible for a text and linguistic analysis on the input text. Text analysis transcribes the input text into a phonetic representation that identifies the units to be synthesized [3] [8]. Linguistic analysis is responsible for transcribing the input text into a linguistic representation such as word pronunciation and lexical stress [3] [8]. The back-end is responsible for generating the acoustic waveform using the information gained from the front-end [6].

The Festival Speech Synthesis System[1] (described in detail in Chapter 3) is a well known, open source concatenative speech synthesis engine that serves as a workbench for the development of TTS systems in different languages [9]. The two techniques of conncatenative synthesis shown in Figure 1.1 are both synthesis techniques of Festival, and each technique produces TTS systems with its own qualities. The technique of diphone concatenative synthesis (DCS) uses the diphones[2] of a particular language as speech units to be concatenated, and has the ability to produce TTS systems that are flexible, but often lacking in understandability, naturalness and pleasantness [5] [10]. Unit Selection Synthesis

[1] A speech synthesis engine designed by the Centre for Speech Technology and Research (CSTR), University of Edinburgh, www.csrt.ed.ac.uk/projects/festival [9]

[2] The phoneme-to-phoneme transitions for a particular language [6]

(USS) on the other hand uses different lengths of pre-recorded speech extracted from a natural speech database as the speech units for concatenation [11]. USS has been proven to produce TTS systems with a high degree of understandability, naturalness and pleasantness [12]. There are two techniques of USS within the framework of Festival. The first is called limited domain (Ldom) unit selection synthesis which has a restricted vocabulary. The technique produces synthesis with a very high voice quality of the words with-in a limited domain [5] [13]. The second technique is called open domain (Odom) unit selection synthesis which uses the sub-word units of a word (diphones, phones or syllables) as the speech units for synthesis [14]. The advantage of this technique is that it can produce TTS systems with the same flexibility as a DCS system and with a similar voice quality as an Ldom system.

This thesis investigates each of the concatenative speech synthesis techniques of the Festival speech synthesis system in the aim of building an advanced TTS for Afrikaans. A hybrid approach to TTS synthesis is also made that combines the advantages of the different techniques.

1.2 Afrikaans and Text-To-Speech synthesis in Afrikaans

Afrikaans is one of the eleven official languages of South Africa. Previous efforts for Afrikaans TTS systems have been made. These systems however do not meet all the requirements of an advanced system. This section gives an introduction to Afrikaans, discusses the motivation for the Afrikaans TTS system, shows the Afrikaans phonetic transcription symbols and discusses previous Afrikaans TTS systems.

1.2.1 Introduction to the Afrikaans language

According to CENSUS 2001 Afrikaans is the home language to approximately 14% of the people in South Africa [15]. The language originates from 17th century Dutch, and is influenced by English, Malay, German, Portuguese, French and other African languages. Together with English it first became an official language in 1925 according to *Act 8 of 1925* [16]. The language was promoted during the years of apartheid and played a major role in white minority rule in Apartheid South Africa. It was later renamed as one of the eleven official languages with the constitution of 1996. The language has different varieties such as

Eastern Cape Afrikaans (later became standard Afrikaans), Cape Afrikaans (as spoken by the coloured people of the Western Cape) and Orange River Afrikaans.

1.2.2 Motivation for an Afrikaans TTS system

The need for an Afrikaans TTS system comes with the growing interest in integrating speech technology into the eleven official languages of the country. South Africa is currently faced with the problems that the majorities of the citizens are not fluent in English and are also not computer literate. Figure 1.3 shows that English is the home language to only 8% of the population. The accompanied language atlas shows that no where in the country is English spoken as the dominant home language.

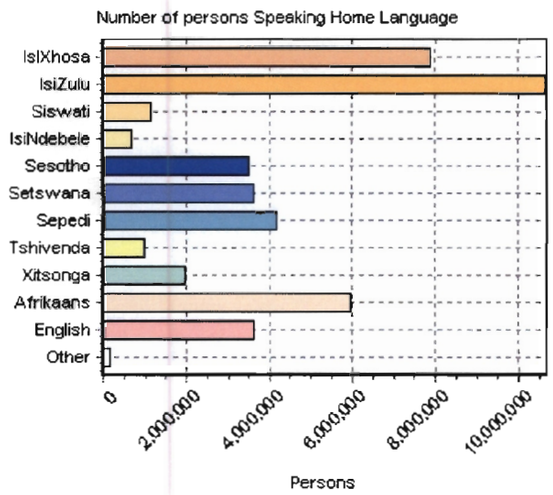


Figure 1.3: South African language statistics [15]

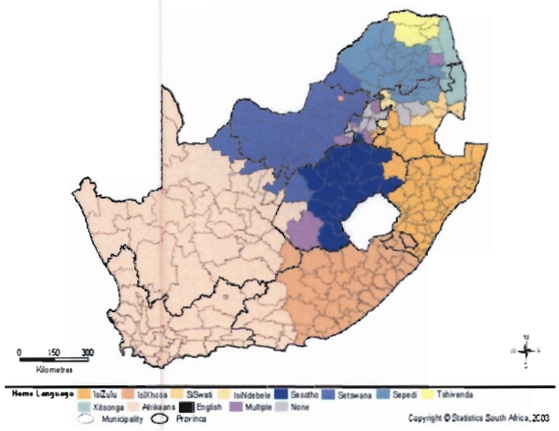


Figure 1.4: South Africa language atlas [15]

The majority of the modern technological systems in the country operate in English. This introduces a problem since the majority of the citizens firstly has difficulties in understanding the systems, and secondly has phobias of using computerized system. These problems build a barrier between technology and the technologically challenged citizens of the country. The aim of integrating speech technology into the eleven official languages is to eliminate these problems by developing and implementing multilingual technological systems that all users can understand and relate to. An example of such a system would be an information access point where a user can simply walk up to a machine, tell the system what information is required in his/her home language, and have the system speak out the information in the same language. The use of such systems would mean that people would now be able to use technological systems simply by communicating with the systems in their mother tongues. Therefore, people will now be more comfortable with using technological systems even with limited or no computer literacy.

Building an Afrikaans TTS system is the first step forward in building the first completely multilingual TTS system for all South African languages. The major aim of the multilingual system is to make technological systems speak all the eleven languages. Such a system has numerous applications that include the field of telecommunications, building devices to assist the visual and hearing impaired, language education and even entertainment. Full explanations of each application are discussed in Chapter 2.

1.2.3 The Afrikaans phonetic transcription symbols

The speech units used for concatenative synthesis are represented by the phonetic transcription symbols of the language the system is intended for. Therefore, the phonetic transcription symbols of the Afrikaans language must be known. The phonetic transcription symbols used for this research are based on the SAMPA[3] phonetic transcription symbols as used by the sub-project "*Korpus Gesproke Afrikaans*[4]" of the "*Protokol vir Breë Fonetiese Transkripsie vir Afrikaans*" project [19]. The aim of this project was to find a controlled phonetic transcription set of spoken Afrikaans, using the SAMPA (Speech Assessment

[3] SAMPA – Speech Assessment Methods Phonetic Alphabet www.phon.ucl.ac.uk/home/sampa/home.htm [17]

[4] Spoken Afrikaans Language Resource [18]

Methods Phonetic Alphabet) Afrikaans character set [20]. Table 1-1, influenced by [19] and [20], shows the list of the phonetic transcription symbols of the Afrikaans language used in this research. Note that double consonants such as "*p-p*" for example are not given definition since they are pronounced the same as the single consonant sound.

Table 1-1: Phonetic transcriptions for Afrikaans

Class	Character	Modified Phonetic symbol	Example within Afrikaans word	Phonetic transcription	English Translation
Plosive consonants	p	p	aap	ap	ape
	b	b	bed	bEd	bed
	t	t	tien	tin	ten
	d	d	dak	dAk	roof
	k	k	kant	kAnt	side
	gh	g	gholf	gOlf	golf
Fricative consonants	f	f	fiets	fits	bike
	w	v	water	vatIr	water
	s	s	sout	s$d	salt
	z	z	zebra	zebrA	zebra
	sj	S	sjampoe	SAmpu	shampoo
	g	x	gras	xrAs	grass
	h	h	huil	h%l	cry
Nasals	ng	N	sing	sIN	sing
	m	m	mens	mEns	human
	n	n	nat	nAt	wet
	nj	J	sjampanje	SAmpANI	Champagne
	l	l	laat	lat	late
	r	r	rol	rOl	roll
	j	j	jas	jAs	jacket
Short vowels	i	I	pit	pIt	pip
	e	E	met	mEt	with
	a	A	mat	mAt	carpet
	o	O	skop	skOp	kick
	u	U	put	pUt	well
Long vowels	ie	i	skiet	skit	shoot
	uu	y	buur	byr	beer
	ee	e	nee	ne	no
	eu	2	deur	d2r	door
	aa	a	skaap	skap	sheep
	oo	o	oop	op	open
	oe	u	soek	suk	seek

	y	&	sny	sn&	cut
	ui	%	trui	tr%	jersey
	ou	$	oud	$d	old
Diphthongs	aai	Q	laai	lQ	drawer
	ooi	W	mooi	mW	pretty
	eeu	B	sneeu	snB	snow

1.2.4 Previous Afrikaans TTS systems

In the year 1995 Mr. M Johan Wagener from the department of computer science at the University of Port Elizabeth did a Doctoral thesis on synthesizing speech from Afrikaans text. The title of the project was "*Synthesising intelligent speech from Afrikaans Text*" [21].

Another effort was made by the speech technology research group at the University of Stellenbosch, South Africa. Their system is used within a hotel reservation booking system, and is called the "*African Speech Technology Project*" (AST) [22]. The system was implemented using the limited domain unit selection synthesis technique of the Festival speech synthesis system. The reason for using the limited domain approach was because it could produce the high voice quality synthetic speech that is natural and understandable enough to be used for telephone applications [22]. The system is however restricted to the specified database of the booking system, and can hence not synthesize any word outside of this domain. This was not a problem for the AST since they only needed to synthesize words within the booking system. Their complete AST system works in Afrikaans, English, Xhosa and isiZulu [22].

1.3 Problem statement

Current Afrikaans TTS systems do not live up to all the requirements of an advanced TTS system. These requirements are:

1. Understandability - a measure of how well a synthesized message is understood by a listener after the first time of listening.
2. Flexibility - the ability of the system to synthesize or attempt to synthesize any possible linguistic entry to the system.

3. Naturalness - a measure of how well the synthetic voice compares to that of real human voice.
4. Pleasantness - this is a measure of how pleasant the synthetic voice is to listen to, which states whether or not a listener will be willing to listen to the system again.

The concatenative synthesis techniques of the Festival speech synthesis system each has its own advantages and disadvantages, and each technique has problems in building TTS systems that meet all the requirements of an advanced TTS system. The technique of diphone concatenative synthesis can produce the flexibility that is required but not the understandability, naturalness, and pleasantness that is also required. Limited domain unit selection synthesis can produce the most understandable, natural and pleasant synthetic speech but at the expense of flexibility. Open domain unit selection synthesis overcomes the problems that diphone concatenative synthesis and limited domain synthesis are faced with by being able to produce understandable, natural, pleasant and flexible synthesis systems, but are often inconsistent [10].

1.4 Objectives of research

The main objectives are to investigate the concatenative speech synthesis techniques of the Festival speech synthesis system and to implement each technique in the design of an advanced TTS system for Afrikaans. The different tasks in achieving this goal are discussed in this section.

The tasks are to:

- Study and understand the general concepts, purpose and implementations of speech synthesis. This includes the history of speech synthesis, how these systems work, how each technique is linked to the theory of the human speech production system, what these systems are used for and finally the different techniques used in building TTS systems.
- To investigate and implement one particular technique called concatenative speech synthesis.

- To investigate and master the different synthesis techniques of the popular speech synthesis system called the Festival speech synthesis system, which is based on concatenative speech synthesis.
- To implement an Afrikaans TTS system using each of the different techniques of the Festival system.
- To test and evaluate each system using subjective listening tests, to show how well each system measures up to the requirements.
- To use the evaluation of the results to propose and build a single system that meets the requirements of an advanced TTS system for Afrikaans.
- To draw conclusions and to make recommendations for future work.

1.5 Scope and limitations

The scope of this thesis is on concatenative speech synthesis, and using this technique in the design of an advanced TTS system for Afrikaans. We concentrate on the concatenative speech synthesis system, Festival, and how its synthesis techniques are used to accomplish the above mentioned task.

Both synthesis techniques, diphone concatenative synthesis (DCS) and unit selection synthesis (USS) were fully investigated and implemented. One DCS system, one Ldom USS, two Odom USS systems and two hybrid systems (Ldom/DCS and Ldom/Odom) were implemented.

From a system's design point of view the limitations were as follows:

- Each system was built in a testing environment since a profession studio was not available.
- No professional recording equipment was used.
- A non-professional speaker was used to record the speech database for each system.

Another limitation came in the form of the time taken to perform manual hand labeling of the speech databases for the DCS and Ldom systems. The labeling of the speech database is a crucial step in the voice building process since it labels the positions of the speech units to be used for synthesis [23]. The accuracy of these labels play a major role in the quality of a TTS

system since incorrect labeling will result in the wrong region of the speech units being used at synthesis. There are automatic techniques for achieving this task, and Festival provides two algorithms for doing so (each discussed in Chapter 3). Results of these algorithms were however not completely satisfactory and therefore hand correction of these labels had to be done.

1.6 Contribution of study

This thesis contributes to the fields of Speech Technology (speech synthesis) and Human Language Technology (HLT) in South Africa. In the field of speech synthesis this thesis contributes a single technique that can be used for the design of an advanced TTS system for any language. It also provides a good overview of the Festival speech synthesis system and the techniques used for building new TTS systems in different languages.

In the field of HLT this thesis contributes an understandable, fully flexible, natural and pleasant Afrikaans TTS system. The technology used in designing this system can also be extended to designing multilingual TTS systems for all the eleven official languages of South Africa.

1.7 Plan of development

Chapter 2 of this thesis is a literature review of speech synthesis fundamentals. It discusses the following:

- The history of speech synthesis
- The human speech production system
- Speech synthesis techniques
- Applications of speech synthesis

Chapter 3 is a detailed discussion of the Festival Speech Synthesis System. It gives a detailed overview of the system, discusses its architecture and the hardware and software requirements. It secondly gives a detailed discussion of each of the synthesis techniques available with the system.

Chapter 4 is a detailed methodology of the steps involved in implementing an Afrikaans TTS system using each of the techniques discussed in Chapter 3. The DCS Afrikaans system is discussed first, followed by the Ldom USS system. The combination of the two using a technique within the framework of Festival into one hybrid Afrikaans TTS system is then discussed. This is followed by the construction of two Odom USS systems using two different speech labeling algorithms.

Chapter 5 presents the results and the evaluations of each of the systems implemented. The subjective test using ten listening subjects and six test sentences are discussed in detail. This includes how the listening subjects were selected and the evaluation sheet designed to evaluate the performance of each system. The results are in the form of tables and graphs. Each system is then evaluated using the results and the feedback gained from each listening subjects, and is further compared to propose one technique to be used for the implementation of the advanced TTS system for Afrikaans. The proposed technique is a combination of the Ldom and the best performing Odom system into a new hybrid system. The resulting system is then implemented as the advanced TTS system for Afrikaans.

Chapter 6 gives a summary of all the work done in this thesis in the aim of achieving the objectives set out in this chapter. This chapter also draws conclusions based on the results obtained in Chapter 5 and makes recommendations regarding the one technique to be used in the implementation of an advanced TTS system for Afrikaans. Directions for future work are also given.

1.8 Summary

In this chapter we introduce the topic of speech synthesis and we also introduce and discuss the need and motivation for an Afrikaans TTS system. We discuss the problems with current Afrikaans TTS system in terms of their abilities to meet the requirements of an advanced TTS system. The objectives for achieving the goal of designing an advanced TTS system for Afrikaans are discussed. The chapter is concluded with the plan of development of this thesis.

CHAPTER 2

Speech Synthesis Fundamentals

It is important to know the origins of speech synthesis, how it works, how it is implemented and its areas of applications. Speech synthesis fundamentals are the foundation for which the current speech synthesis techniques are based on and it is therefore worth looking at the history of this technology. The aim of speech synthesis is to mimic or imitate the human speech production system and therefore also needs to be discussed. There are three major techniques used for the implementation of speech synthesis systems. Each technique is discussed in this chapter with an emphasis on the techniques of concatenative speech synthesis. Finally the deliverables of speech synthesis systems are discussed in terms of its applications for the real world.

2.1 History of Speech Synthesis

The history speech synthesis dates back to the 17^{th} century [4]. In 1779 a Russian Professor by the name of Christian Kratzenstein used acoustic resonators activated by vibrating reeds to produce five long vowels sounds (/a/ /e/ /i/ /o/ and /u/) [3]. The aim of this project was to explain the physiological differences between the vowels [3]. This was the beginning of mechanical speech synthesizers. Figure 2.1 shows the resonators designed for each of the five vowels.

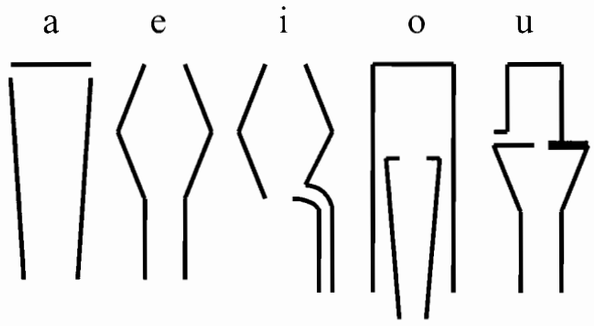


Figure 2.1: Kratzenstein's acoustic resonators [3]

Later more systems were introduced by Mr. Wolfgang von Kempelen in Vienna 1791 and Mr. Charles Wheatstone in the mid 1800's. Wolfgang von Kempelen's system, called the "*Acoustic Mechanical Speech Synthesizer*" had the ability to produce single and combinational sounds [3] [24]. The system consisted of a pressure chamber to act as the lungs, a vibrating reed to act as the vocal cords and made use of a leather tube for vocal tract action [3]. Vowels were produced by manipulating the leather tube while consonants were produced by manipulating four separate constricted passages controlled by the fingers [3]. A reconstruction of Mr. von Kempelen's system as made by Sir Charles Wheatstone in the 18th century is shown in Figure 2.2.

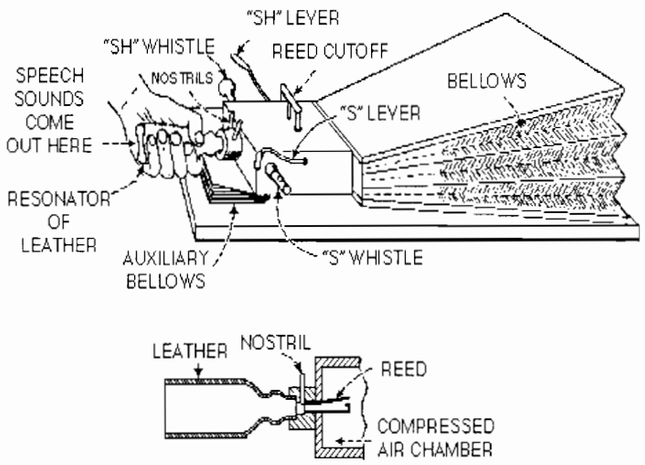


Figure 2.2: Reconstruction of Wolfgang von Kompelen's Acoustic Mechanical Speech Synthesizer [3] [26]

The first practical device to be considered a speech synthesizer was built by Homer Dudley in the mid 1930's [3]. The system was called the VODER (Voice Operating Demonstrator). It was first introduced and demonstrated at the World's Fair in New York and in San Francisco in 1939. Another breakthrough practical system was introduced by the United Kingdom Telephone Company when they introduced their speaking clock in 1936 [7] [25]. The system could literally tell the time by concatenating phrases, words and part-words from an optical storage space. Work on the VODER system started at the same time as the speaking clock was built [3]. The initial device was called the VOCODER (VOICE CODER) and later became the inspiration for the VODER system [3]. The VOCODER analyzed speech into varying parameters that drove a synthesizer to do the reconstruction of the approximated speech signal [3]. The VODER system, shown in Figure 2.3, made use of a wrist bar for selecting a voice or noise source, and used a foot pedal for selecting the fundamental frequency of the output [26]. The voice or noise source was then passed through ten bandpass filters, operated by the fingers to produce the final output. The system performed poorly in terms of the resulting speech quality, but it was this potential that led to more interest in building high-quality speech synthesizers [26].

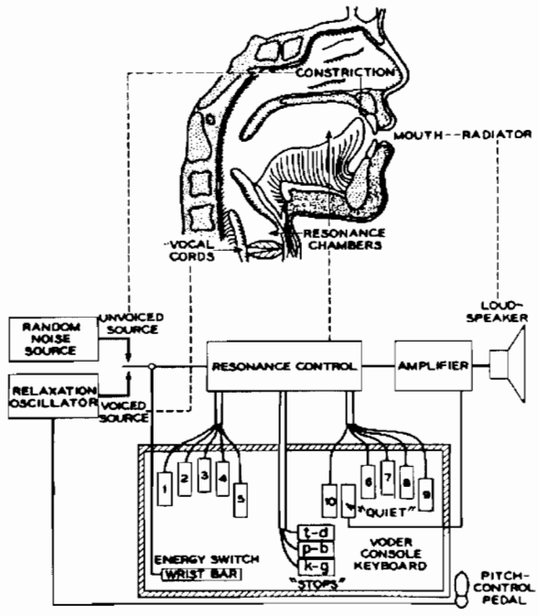


Figure 2.3: The VODER speech synthesizer [3] [27]

open space between the upper end of the *trachea* and between the *vocal cords*. The *larynx* forms and air passage to the lungs and it holds the *vocal cords* [33]. The e*piglottis* is a cartilage at the root of the tongue that protects the *trachea* from food and liquid in the swallowing process [3]. When swallowing the cartilage is depressed so that whatever is being swallowed can not enter the windpipe. The *pharynx,* also known as the throat, is the lined cavity behind the mouth and the nose. The *velum*, also known as the soft plate, controls the air flow from the *pharynx* to the *oral* and *nasal cavities*. The *oral cavity* is the main source of speech output and has changeable dimensions [26]. The dimensions are shaped by the movements of the lips, tongue and the velum. The *nasal cavity* has a fixed shape and dimension and is responsible for producing the nasalized sounds of speech [26]. Figure 2.3 is a schematic block diagram of Figure 2.4, and is used to describe the speech production process in a step-by-step manner.

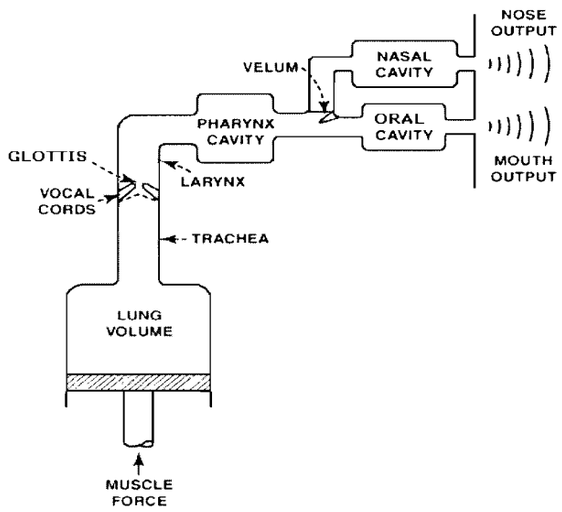


Figure 2.5: Systematic block diagram of the human speech production system [33]

The "*source filter theory of speech production* [34]*"* states "*acoustic human speech is the results of a combination of a source of sound energy modulated by a filter determined by the vocal tract*". The speech process starts at the lungs and the diaphragm which serves as the main energy source for producing the air flow into the vocal tract (muscle force as shown in Figure 2.5). When air flow passes from the *trachea* to the *larynx* the *vocal cords* and the *glottis* respond differently depending on the sound that is going to be produced. When the *vocal cords* vibrate during this process voiced sounds (vowels and

voiced consonants) are produced. When the *vocal cords* are open at the time when air flows through the *glottis* unvoiced sounds are produced [26]. When the *vocal cords* suddenly open up from a completely closed position it causes a stop consonant sound. Nasal sounds are produced when the *velum* is lowered, connecting the *vocal tract* to the *nasal tract* (area from the *velum* to the *nostrils*). The fundamental frequencies of the vibrations of the vocal cords differ for men, women and children. The average fundamental frequencies are; 110Hz for men, 200Hz for women and 300Hz for children [3].

Different techniques and theories exist to mimic or reproduce the speech produced by the human speech production system. Each technique is aimed at modeling the production system differently and has its advantages and disadvantages. These techniques are discussed in the following section with an emphasis on concatenative speech synthesis. This technique does not model or attempt to model any part of the human speech production system since it uses the actual output of the human speech production system.

2.3 Speech Synthesis Techniques

The modern speech synthesis techniques discussed in this section each have their own unique way of producing artificial speech. These techniques can be seen as modern versions of the techniques used in the past since they work on the same basic principles, but use modern technology as their engines. The techniques of articulatory speech synthesis, formant speech synthesis and concatenative speech synthesis are discussed in this section with on emphasis on the latter.

2.3.1 Articulatory Speech Synthesis

Articulatory synthesis is based on modeling the human vocal organs and the simulation of the resonance effects of the vocal tract (area from the glottis to the oral cavity, see Figure 2.4) [3] [6] [8]. Because this method is aimed directly at the fundamental theory of how we as humans produce speech, it should produce the most satisfying synthetic speech [3]. However, this is not the case. The data used by this technique is acquired from X-Ray data of a human producing natural speech. This data is two-dimensional,

whereas the real vocal tract is three-dimensional. Therefore the loss of information or rather insufficient amount of information leads to short comings of articulatory synthesis [3]. With the introduction of MRI[5] (Magnetic Resonance Imaging) data this problem can be overcome [28]. With MRI, high-resolution three-dimensional shape data is available and hence the short comings two dimensional data are overcome. The drawbacks of the MRI technique is that the image acquisition periods are long, and is it complicated to make the teeth and bones appear different from air [28]. Research is currently going into overcoming these drawbacks, and currently the image acquisition periods are becoming less [28].

With more knowledge of the vocal tract, vocal tract imaging during speech, aero acoustics, speech disorders, singing and emotional states, articulatory synthesis has the potential of being the preferred method of producing artificial speech in the future [3] [28].

2.3.2 Formant Synthesis

This technique is based on modeling the formant frequencies produced by the human speech production system [6] [26]. The advantage of this technique is that it provides an infinite amount of sounds that are available for synthesis and for this reason systems built using this technique will have more flexibility than systems built using other techniques [6]. Formant synthesis is based on the source-filter model of speech which is shown in Figure 2.6.

The goal of the source-filter model is to replicate natural speech by using model parameters extracted from the natural speech [26]. The model parameters used here are the excitation and vocal tract parameters. Voiced sounds are represented by a periodic impulse train generated by an impulse train generator. Unvoiced sounds are represented by a white noise signal which is generated by a random noise generator. Both excitation signals are then scaled down, and a voicing switch is used to select between the two. For combinational sounds of both a voiced and an unvoiced excitation, a mixed excitation rule is used to properly scale and sum the two signals to produce the combinational sound

[5] MRI - an imaging technique used to produce high quality images of the inside of the human body [35]

[26]. The glottis signal is generated once all the excitation signals are simulated. This is followed by the vocal tract filter modulation of the excitation modulation within the vocal tract, and finally speech is produced.

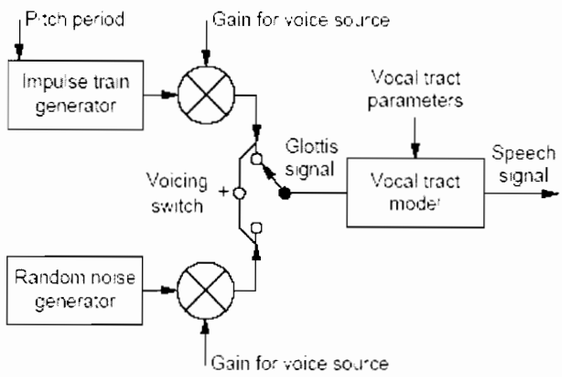


Figure 2.6: The source-filter model of speech production [26]

Cascade Synthesis and Parallel Synthesis are two techniques of Formant synthesis used for producing synthetic speech [3]. Cascade synthesis consists of a string of band-pass filters connected in series as shown in Figure 2.7. The output of each resonator is the input to the next, and only the formant frequencies are needed as control information for this technique. With this comes the advantage that the formant amplitudes for vowels do not need individual controls. The disadvantage of cascade synthesis is that it only works well for non-nasal voiced sounds, but the generation of fricatives and plosives are a problem [3].

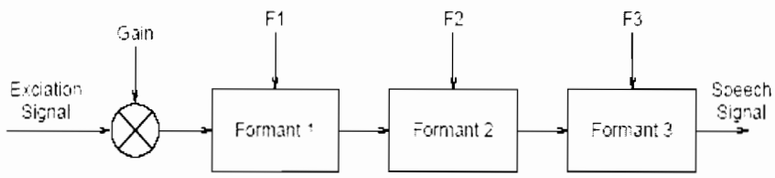


Figure 2.7: Block diagram structure of Cascade Formant Synthesis [3]

Parallel synthesis consists of resonators connected in parallel. As shown in Figure 2.8, the excitation signal is applied to all of the formant resonators simultaneously, and outputs are then individually gained to improve the quality of the resulting speech [3]. The advantage of parallel synthesis is that it works well for nasals, fricatives and stops [26].

However, the technique can not model some vowels as well as the technique of cascade synthesis [3].

Chapter 2. Speech Synthesis Fundamentals

The first fully automatic synthesizer was based on articulatory synthesis [28]. In the year 1958 George Rosen of the Massachusetts Institute of Technology, MIT, introduced the world to articulatory synthesis with the DAVO (Dynamic Analog of Vocal tract) speech synthesizer [24] [29]. Up until today articulatory synthesis systems are still being development to act as research into the fundamental aspects of speech production, and are believed to outperform concatenative speech synthesis in the future [28].

The first speech synthesis systems considered to be intelligent enough to be used for commercial use was based on formant synthesis [28]. Formant synthesis was introduced to the world by Walter Lawrence in the shape of his PAT (Parametric Artificial Talker) speech synthesizer in 1953 [3] [24]. The synthesizer consisted of a parallel circuit of three electronic formant resonators, to which the input was either a buzz or a noise. It was operated using a glass slide that converted painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency and the noise amplitude [3] [30]. The slide was scanned with what was called a "*flying top*" cathode ray tube light source and a photo-electric cell. This created the formant waveforms by heterodyning with a resultant amplitude modulated waveform of 10 KHz.

Work on concatenative synthesis started in the 1970's, but these systems only became more practical when computer hardware became cheaper and more robust [7]. In 1974 Ignatius Mattingly [31] stated that "*The advantage of a simulation (by computer) is that it can be completely reliable and accurate, and the design of the synthesizer can be readily modified; the disadvantage is that an extremely powerful computer is required and such computers are too expensive to permit extended real-time operation.*" With the state of modern technology and the realization of the so called "*super computer*", this is no longer the case. With faster computers and larger disk space being more accessible, modern techniques of concatenative synthesis can be used to exploit the power of the computer simulation spoken of by Ignatius Mattingly.

At turn of the 1980's, Yoshinori Sagisaka of the Advanced Telecommunications Research unit developed a system called *nuu-talk* [7] [32]. The system was able to use a large inventory of speech units for synthesis, and Sagiska made full use of this. Instead of having one example of each speech unit to choose from, he had access to variations of the

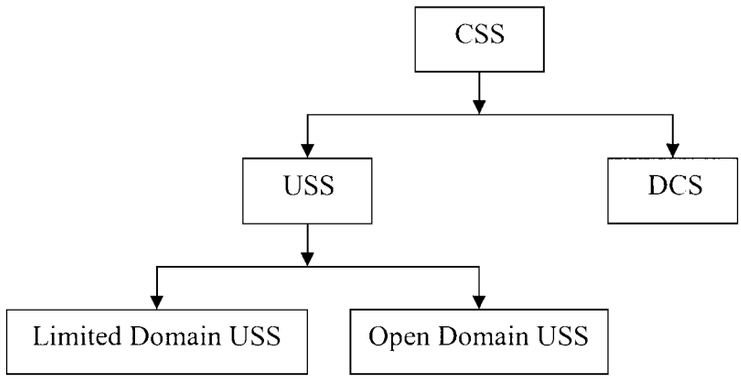


Figure 2.9: Techniques of Concatenative Speech Synthesis

Diphone Concatenative Synthesis

Diphone concatenative synthesis uses the diphones of a particular language as the speech units for concatenation and synthesis. Diphones are simply the phone-to-phone transitions, from the middle (steady state region) of the first phone to the middle of the second phone [6] [7]. The transitions contain the adjacent points between two phones, which mean that the concatenation points from diphone to diphone are in the steady state regions of each phone. The result of this is a reduction in concatenation distortion between the concatenation points of two diphones [3]. The number of diphones present in a language is the square of the number of phones in the language [6], which gives a really good coverage of all possible sounds present in the language. For this reason flexibility is a big advantage of a diphone concatenative speech synthesis system [5]. Another advantage of DCS is that the memory requirements of the diphone database are not as demanding as the requirements for systems based on larger units. A disadvantage of this technique comes in the form of audible discontinuities at the concatenation points of diphones [30]. Despite these drawbacks a number of systems have been built using this technique including one by the AT&T Bell Laboratories [40].

Unit Selection Synthesis

"Unit selection synthesis, where appropriate sub-word units are selected from multiple examples in a database of natural speech, has been shown to produce high quality

natural sounding speech", [41]. The major difference between USS and DCS is that USS uses a more general speech database with more examples of each speech unit available for synthesis. DCS uses a monotone speech database with only one example of each speech unit available for synthesis while USS usually has more than two.

The sizes of the speech units can be varied from phones, to diphones, triphones or even full words [42]. As shown in Figure 2.9 two forms of USS exist, limited domain USS and open domain USS. A limited domain system (Ldom) uses full words as the units for synthesis, and an open domain system (Odom) uses smaller sub-word units such as phones and diphones. The choice of the unit size also produces different results. For example, a full word Ldom system will have a more natural sounding voice than an Odom system that uses diphones. A full words system will however not have the flexibility of the diphone system, so a compromise has to made keeping in mind that the unit size chosen will determine the overall quality of the system. To select the appropriate unit for synthesis USS makes use of a unit selection mechanism to determine and select the appropriate unit at run-time [10] [12] [43]. The speech database used for USS is acquired by recording natural speech in the form of sentences which have the necessary prosodic information of the natural speech that is needed to make the resulting system sound as natural as possible [12].

As mentioned USS makes use of a unit selection mechanism to select the appropriate unit for synthesis, this is called *Run-Time Unit Selection* [43]. A very flexible and trusted system called CHATR [10] has been used for selecting units from range of databases, including male and female speakers [12] [44]. The fundamental unit size used by CHATR is phone. CHATR employs two cost functions (shown in Figure 2.10) to determine and select the appropriate unit for synthesis. The first cost function called is the *target cost,* $C^t(t_i , u_i)$ is used to determine how well each database unit, (u_i), matches an ideal target unit (t_i). The second cost function is called the *concatenation cost* or *joint cost,* $C^c(u_{i-1},u_i)$ is an estimate of the quality a possible join between phone units. The units with the least target cost and join costs are the units selected for synthesis.

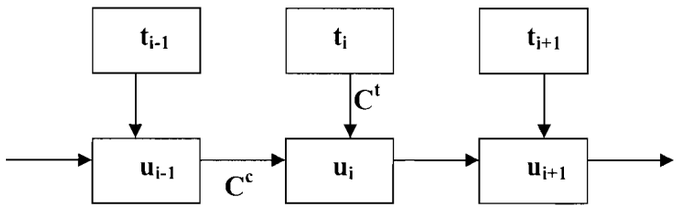


Figure 2.10: Cost of unit selection in unit selection synthesis [12] [43]

Each target phone and each candidate in the database is typified by a multidimensional feature vector [43]. The features include pitch, duration, stress, power and also hold phonetic context and prosodic characteristics of the preceding and following phones. The *target cost* is determined by the weighted sum of the differences between the elements of the target feature vectors and candidate feature vectors [12]. The differences between the elements are known as the *p target sub-costs* given by $C^t_j(t_i, u_i)$, where $j=1,....,p$. The *target cost,* given weights w^t_j for the sub-costs, is then calculated by equation 2.1 shown below:

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w^t_j C^t_j(t_i, u_i) \tag{2.1}$$

The *concatenation cost* is also settled by the weighted sum of *q concatenation sub-costs,* given by $C^c_j(u_{i-1},u_i)$, where $j=1,..., q$. These sub-costs can be determined by the characteristics of the candidate unit u_i and the previous unit u_{i-1}, but can in addition be derived by the signal processing of the units [12]. The *concatenation cost,* given weights w^c, for the sub-costs, is then calculated by equation 2.2 shown below:

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w^c_j C^c_j(u_{i-1}, u_i) \tag{2.2}$$

The total unit selection cost for the CHATR unit selection mechanism is then the sum of the *target* costs and *concatenation* costs for *n* units in an utterance. It is also important to note that two additional factors must be added. These factors are the transitions from the initial silence to the first phone and the transitions from last phone to the concluding silence in an utterance [43].

The final equation for the total unit selection cost is then:

$$C(t^n_1, u^n_1) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad (2.3)$$

Where $C^c(S,u_1)$, defines the concatenation cost of the transition from the initial silence (S) to first phone unit, and $C^c(S,u_n)$ is the concatenation cost of the transition from the last phone unit to the concluding silence (S). The goal of the CHATR mechanism is to use the phone unit with the minimum *target* and *concatenation* costs [12] [43]. This cost is determined by:

$$\hat{u}^n_1 = \min_{u_1,\ldots,u_\alpha} C(t^n_1, u^n_1) \quad (2.4)$$

The Viterbi search described in [10] is used to perform optimal unit selection that can obtain near real time synthesis on a large database, provided the search is pruned. The authors of [10] produced a new or modified technique for robust unit selection from a large database of units. Instead of using the fundamental units of phones, they used half-phones. These units are equivalent to half diphones and allow a combination of these units to imitate units like diphones and phones [10]. For CHATR this also means that there will be instances where the system profits from a diphone-type join as opposed to a phone-phone joint and hence potentially produce better quality synthesis. The results obtained by [10] shows that with a combination of a high-quality database, and their improved unit selection technique, better overall quality synthetic speech can be achieved.

Recently researchers in the field of speech synthesis have been looking at optimal database design techniques for building unit selection systems [14] [41]. These systems make use of an optimized database with a very good coverage of units with the appropriate prosody needed to produce high-quality speech synthesis. These databases are smaller than general databases and therefore have faster synthesis times [12].

There are many commercial systems that use the technique of USS simply because it can provide the high-quality speech synthesis that is needed for their applications. The authors of [28] states the following "*We use concatenative synthesis because that is*

currently the best available method to produce synthetic speech of consistently high qualit however, at the same time we also believe that in the long run concatenative synthesis is not the answer". The reason for the latter part of this statement is because even though this technique provides what is needed for advanced TTS systems, it still has fundamental limitations. Fundamentally concatenative speech synthesis does not reproduce or imitate the human speech production system. It instead replicates the speech produced and is therefore limited to a particular speaker and predefined databases. Unlike the other techniques of speech synthesis, concatenative speech synthesis does not have the freedom that comes with modeling the human speech production system, even though it produces better results.

2.4 Applications of speech synthesis

Verbal speech is the dominant form of communication between humans. When people communicate via speech it is easy to analyze, interpret and understand the message being conversed. Extending this form of communication to computers opens new doors for verbal communication between humans and machines. The applications of speech synthesis are very open since the main purpose is to read text out loud. The application a TTS system is intended for is usually dependant on the synthesis technique the system is based on. For example it would be better to use limited domain techniques for applications that need a limited vocabulary (example menu systems, speaking clocks). And it will be better to use open domain techniques for applications with an open vocabulary like electronic books, and full TTS systems for specific languages.

2.4.1 Aid for the visual and hearing impaired

Speech synthesis can provide communication assistance for the visual and hearing impaired. Blind people can not see and therefore are not able to read books, newspapers, browse the internet for information, read electronic mails or retrieve any textual based information. They are limited to the use of their hearing and speaking abilities to transmit and retrieve information. A speech synthesis system can therefore help by filling this gap in the communication strategies used by the visually impaired. Systems such as reading machines can allow blind people to browse the internet, read electronics mails and

basically retrieve any information that is in the form of text. Before speech synthesis reading machines used audio recordings in the form of tapes to read textual information such as books [3]. This was a very expensive exercise since it usually took several months to record the books onto tapes. Modern speech synthesis systems have overcome this problem by producing speech from text at the time the information is needed.

The hearing impaired can not hear, and they also have speech imperfections. Even though they can see, which enables them to read and write, they can not communicate audibly. Their most trusted form of communication is the use of sign language. Installing speech synthesis software onto hand held devices such as PDA's[6] will allow the deaf to use these devices to audibly communicate with able bodied people. This is done by typing the desired message into the device, which then reads out the message to the listener.

2.4.2 Telecommunications

In the field of telecommunications speech synthesis systems can be used in telephone enquiry systems, telephone booking systems, telephone relay services and Interactive Voice Response, IVR systems to name a few [45]. One of the most exciting applications for this technology is that of a talking SMS[7] service for cellular phones (now also available on land line phones in South Africa). SMS has become a major asset for cellular phone operators due to its popularity amongst the youth of South Africa. With the growing interest of building advanced TTS systems for the eleven official languages of this country, the cellular phone companies including the land line service provider could soon have talking SMS systems in all of the eleven official languages. This will enable SMS users to send an SMS in their preferred language and also have it read out loud in the same language. These systems have yet to be implemented but will hopefully appear in the near future if more research is done.

[6] PDA – Personal Digital Assistant

[7] SMS – Short Message Service

2.4.3 Education and language education

Speech synthesis can play a vital role in the early stages of teaching children to read and write. Children will benefit from this technology by replacing the role of a teacher or tutor at times when the child is away from school, and the system can even act as an assistant to teachers. It is not easy to learn to read and write without the aid of vocal help and a speech synthesizer is the perfect system to provide this form of assistance 24 hours a day, seven days a week. Not only can this technology speed up the process of language education in children, but anyone can now learn a new language without the aid of a tutor. All that is needed is an educational book that teaches the language and a TTS system in the language to assist with the pronunciations of words.

2.4.4 Electronic services

Electronic services such as information access points can benefit from multilingual speech synthesis systems by supplying people with the information needed in their home languages. In South African this is very useful since there are eleven official languages. English is not understood and spoken by the majority of South Africa, and most of our people are not educated to the level that they are completely comfortable with modern technology and computers. Having systems that people can understand, easily use and relate to will make their lives easier by eliminating the phobias of using modern technology.

All the applications of speech synthesis systems mentioned in this section will have a direct impact on South Africa when it is applied to the eleven official languages. Designing the Afrikaans TTS system is the first step to achieving this goal. The technology used for building this system has been extended to TTS systems in some of the other eleven languages and is presented in Chapter 4.

2.5 Summary

The goal of this chapter was to give an over of speech synthesis fundamentals. We discussed the history of speech synthesis, where and when everything started, we then

discussed the human speech production system and how speech synthesis is aimed at producing the same output with the use of modern technology. We then discussed in detail the different techniques used to build speech synthesis systems. Here we discussed articulatory speech synthesis which is aimed at modeling the human speech production system, we then discussed formant speech synthesis which is aimed at modeling the formant frequencies of human speech and we discussed the current state of the art technique called concatenative speech synthesis which uses the direct output of the human speech production system by concatenating units thereof to produce synthetic speech. Some of the applications of speech synthesis were then discussed which provides motivation for the research into speech synthesis technology.

CHAPTER 3

The Festival Speech Synthesis System

The Festival speech synthesis system is a well known, freely available speech synthesis engine that offers a general framework for the development of new speech synthesis systems in different languages [8] [9] [46]. Development of the system started in 1996 at the CSTR (Centre for Speech Technology Research), University of Edinburgh, Scotland. The system works on the basis of concatenative speech synthesis by diphone concatenative synthesis and unit selection synthesis. This chapter gives a general overview of the system and its operation. It also discusses the two synthesis techniques in detail, including the advantages and disadvantages of each technique. A discussion on a hybrid approach that combines the advantages of the two techniques within the Festival framework is also given.

3.1 Introduction to Festival

3.1.1 System overview

The core system and architecture of Festival is written in C++, and a Scheme based command interpreter is used for other Festival modules [47]. The Scheme interpreter (SIOD - Scheme In One Defun 3.0) offers a Lisp interpreter that is suitable for embedding applications like Festival as a scripting language [46]. One advantage of using a Lisp interpreter is that the parameters of the application (example Festival) can be controlled at runtime by the interpreter without having to change the underlying C++

code or recompiling the code which contributes to the power and the accessibility of the system [8] [45].

Festival uses the Edinburgh Speech Tools[8] (EST), a library of speech tools for its low level architecture [46] [48]. This library of tools contain support tools and analysis tools such as pitch trackers, finite-state transducers, classification and regression tree builders, a Viterbi decoder, ngram builders, waveform I/O (Input/Output) tools, which is used by Festival to perform key processes in building new synthesis systems [46] [48].

3.1.2 System architecture

The architecture of the Festival speech synthesis system has been formalized for storing the linguistic data in an utterance (text to be rendered as speech) in such a way that it complies with the following requirements [47]:

- Linguistic representations such as words, phrases, syllables and phones must be possible.
- There must be a mechanism that can identify a set of phones that a certain word comprises of.
- Changes in sub-modules must be localized so that changes made to these models do not affect other modules within the system.
- Information on the linguistic entries to the system must not be made redundant or duplicated. The reason for this is that when a change made to one specific piece of duplicated information it must be made to the other version as well, if not the version will become outdated.
- The architecture should not conform to any particular linguistic theory since Festival is to be used for building multilingual TTS systems which will have varieties of linguistic theories.
- The system is to be used for real-time applications and therefore must be efficient and fast enough to accomplish this task.

[8] http://www.cstr.ed.ac.uk/projects/speech tools/manual-1.2.0/ [48]

Previous speech synthesis systems such as *MITalk* [49] and the CSTR *Alvey* system [50] used architectures based on a *string re-writing* mechanism as its fundamental data structure [47]. This type of structure stores the linguistic representation of an utterance as a string, which is then *re-written* with the addition of extra symbols as processing occurs. The major disadvantage of this technique is that it becomes too complicated to handle utterances with complex linguistic representations including words, phones, stress symbols, phrase symbols etc. Other systems including early versions of Festival used an architecture based on *multi-level data structures* (MLDS). MLDS separates different forms of linguistic information into different *streams* as linear lists or arrays of items, i.e. word streams, phone streams, syllable streams etc. Usually co-indexing between streams is needed since information in the streams is all related [47]. Problems occur here due to the fact that the numbers of streams that are needed to cover all the different forms of linguistic information become high, thus making the representation of linear structures and the co-indexing of items difficult [47].

The architecture of Festival defines the utterance structure of an utterance as a set of linguistic *items* and a collection of linguistic *relations.* Linguistic *items* are the *feature structures* of the utterance units such as words, syllables and phones, while linguistic *relations* are simple structures of *items* such as lists or tress [8] [46]. The major difference between this type of architecture and others such as *string re-writing* and MLDR is that this architecture does not restrict linguistic *items* to singular structures such as linear lists. It allows for any graph structure, example trees, which in the end lead to better representation of the items. For example the relation *SylStructure* is a list of items of which each node represents a syllable in a word. From top to bottom this tree structure consists of the words, the syllables and the segments that make up the syllables. This is shown in Figure 3.1.

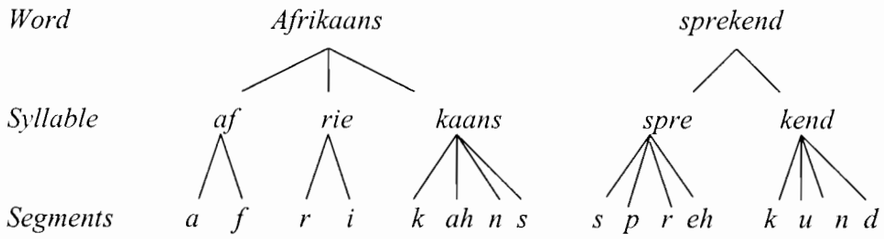


Figure 3.1: *SylStructure* relations between *word* and *segment*

The utterance structure is at the heart of the Festival speech synthesis system's architecture [47]. The linguistic definition of an utterance reads as follows "*an utterance is an uninterrupted chain of spoken or written words not necessarily corresponding to a single or complete grammatical unit*" [39]. The general architecture of an utterance structure as used by Festival consists of the following *relations* [46]:

Utterance The single item representing the input of organized text that is to be converted in to speech. This is at the top of the overall tree structure.

Token This relation is a list of trees of which the roots are the tokens extracted from the input utterance and, the daughters of each root are the *words* associated with each token.

Word This relation is the list of words that make up the input utterance. It is also the leaf nodes of the *token* relation and the *phrase* relation, to be discussed next. Also appears as possible leaf nodes to the *syntax* relation.

Phrase This is a list of trees with phrases as its roots and words as its leaf nodes.

Syntax This is the same as the word relation. The nodes of the word list (syllables) and the terminal nodes of the syntax tree points to the same items

SylStructure This relation links the *Word, Syllable* and *Segment* relations as shown by the example in Figure 3.1

Syllable This is a list of syllables of which the parents are words and the daughters the segments that make up the syllable.

Segment These are a list of phones that are the leaf nodes or daughters of the SylStructure relation.

IntEvent This is a list of intonation events (accents and boundaries) that are related to syllables through the Intonation relation.

Intonation This is a list of trees that relates syllables to intonation events. The roots of these trees are syllables and its leaf nodes are the IntEvents.

Wave This is a single item with a feature whose value is the generated waveform.

The philosophy of the text-to-speech approach by the system is to take the input utterance, through a step by step chain of modules, each adding more information to the utterance structure until the final waveform is generated [46]. This step by step procedure consists of the following components:

Tokenization

This is the first stage in the text analysis phase of the front-end shown in Figure 1.2. A token is an element separated with white space from text or a string, the roots of the token *relation*. Tokenizing according to [46] is the conversion of a string of characters into a list of tokens. The lists of tokens are special characters like numbers and abbreviations that have words associated with them, the daughters of the token *relation*.

Token identification

This step identifies the types of tokens present in the input utterance, example abbreviations and digits such as years, numbers and dates.

Tokens to words

Festival describes a *word* as an element that can be given pronunciation using a lexicon database[9] or a set of letter to sound rules. Tokens that have words associated with them are given definition here.

Part of speech (POS)

This component identifies the syntactic part (grammatical representation) of speech for the words that make up an utterance.

Prosody phrasing

This component makes the synthetic speech of the synthesis system more understandable. In [41] it is explained that prosody concerns the communication of information between a speaker and a listener. Prosodic phrasing in Festival is based on what we as humans do when we use phrasing to mark groups within our sentences. Festival adds punctuation into utterances to perform the task of prosodic phrasing. In [7] it states that it is nearly always the case that punctuation does mark prosodic boundaries, and using a pronunciation based prosodic phrasing algorithm will almost never make a false boundary marking. Festival supports CART trees and full statistical models trained with real speech data as prosodic phrasing algorithms to add prosody to its synthetic speech [7] [46].

[9] A subsystem that provides pronunciation definitions for words

Lexical lookup

This process finds the pronunciations of words from a predefined lexicon or a set of letter to sound (LTS) rules. By definition a lexicon is a dictionary of words, in Festival a lexicon is defined as a subsystem that provides pronunciations for words. The lexicon consists of the following three parts

1. An addenda - usually hand added words.
2. A compiled lexicon - consists of a very large number of words (~10 000 words).
3. Letter to sound rules - gives the system information on the pronunciation of a particular letter.

Each lexicon entry is made up of the following three parts, and an example of the word *Francois* as is shown in Figure 3.2

1. The headword - the text to be synthesized. Might be a word, a number or an abbreviation
2. The part of speech - grammatical representation.
3. The pronunciation - contains the syllable structure, stress markings and the phones that exist in the word.

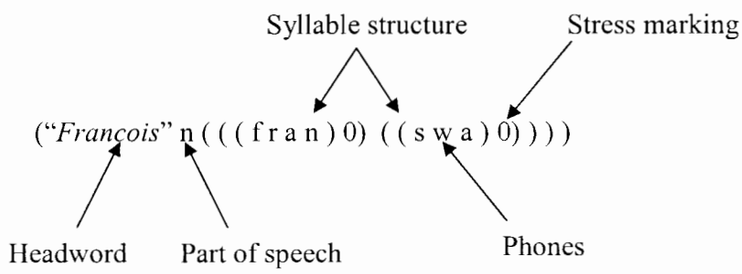


Figure 3.2: Lexicon definition of the word "*Francois*"

The basic approach to the lexical look up process is to search the entire lexicon for the headword. If a match is not found in the addenda then the next step is to look for a match in the compiled lexicon. If the match is not found there, then the LTS rules system is implemented. This system then uses the information regarding the pronunciations of the

phones that make up the head word in order to synthesize the closest possible pronunciation of the word.

Intonation accent

This component adds or assigns accent information to the syllables of the utterance. The Festival modules used to perform this task are; *Default intonation, Simple intonation, Tree intonation, Tilt intonation* and *General Intonation.* The *default intonation* model simply creates target frequencies for the beginning and the end of the utterances. The default frequencies are 130Hz (start) and 110Hz (end). This means that an utterance will start at a frequency of 130Hz and end with a lower tone at 110Hz. These values can however also be changed. The *simple intonation* module uses CART trees to predict whether a syllable is accented or not. Two values are used here, *none* and *hat.* The value *none* means that no accent is predicted for the syllable while *hat* means that the syllable has local maximum accent at the midpoint. The *tree intonation* method uses a linear regression model to predict the start, mid-vowel and end targets for a syllable. The *tilt intonation* module uses the observable information in the F0 (Fundamental frequency) contour of the utterance to predict accents. The *general intonation* module uses the same method as the *simple* module to predict the accents on syllables. Each of these modules is a research question in its own respect and various modules now exist [51].

Assigning duration

This component adds duration information (the time duration, in seconds, that a segment is synthesized for) to each of the segments that make up utterances to be synthesized. Different modules for duration prediction exist within festival of which each is affected by a global duration stretch. The first module called *Default duration* uses durations of 100 milliseconds for each phone in the database. The *Average duration* module calculates and uses the average durations of each segment instead of a fixed duration. The author of [8] states the following "*These two (Average and Default duration) is an obvious over-simplification of what actually happens in natural speech*", this is true since not all phones last for the same fixed duration, and because there is no real consistency in single

phone durations since humans never speak at the same pace. For these reasons the *Klatt duration* model described by [46] uses the minimum and inherent durations together with a set of rules (to allow for modification) to determine the durations of segments. The module is based on the following requirements:

1. Every phone has an inherent duration as one of its distinctive properties.
2. Each rule has the job of increasing or decreasing the duration by a percentage.
3. Each segment has a minimum duration that can not be compressed.

These requirements are formulated into the equation:

$$DUR = MINDUR + \frac{(INHDUR - MINDUR) * PRCNT}{100} \quad (3.1)$$

This module makes use of eleven rules to produce acceptable results, and initial specifications consisting of phone name, inherit duration and minimum duration must be set [8] [46].

The *CART duration* module and has two methods of predicting segment durations. The first method predicts the durations directly for each segment and the second method (known to produce better results [46]) uses *zscores* (the number of standard deviations from the mean durations) instead of predicting the durations directly. The requirements for this method are; the mean duration of each segment in the database and its standard deviation. The duration is then calculated by the formula:

$$Duration = mean + (zscores * standard\ deviation) \quad (3.2)$$

Once acceptable durations of segments are obtained the global stretch parameter can be manipulated to change the pace of the final output.

Generate F0 contour

This component generates the tune of the target utterance from the accent and duration predictions discussed earlier. The methods used for achieving this task are the same as

the methods used for intonation. Intonation is split into two parts. The first predicts accents and the second predicts F0 targets for the target utterance. Each module discussed in the *Intonation accent* section can perform the task of predicting F0 targets, but they do need the information on the durations of the segments [46].

Render waveform

This is the final component in the process of converting text to speech in Festival. This step uses all the information gained by the chain procedure to generate the final waveform for synthesis. Depending on the method used for synthesis, DCS or USS, this component may also take several steps to compute the final waveform [46].

The architecture, the utterance structure and the text to speech conversion process of the Festival speech synthesis system is now known. Next we discuss the system requirements to run Festival and tools required to build new TTS systems within its framework.

3.1.3 System requirements

The version of Festival used for this research is Festival 1.4.3. This section discusses the hardware and software requirements required to run the system and to build new TTS systems within using the system.

Hardware requirements

Festival was built and designed to run in a *UNIX* environment under *Linux*. The system used for this research is Linux 2.4.24, KDE (Knoppix Development Environment) version 3.3.2. Festival has also been compiled and tested under Sun Sparc Solaris (2.5.1, 2.6 and 2.7), SunOS 4.1.3, FreeBSD (2.3, 3.x ELF based), Redhat (4.1, 5.0, 5.1, 5.2, 6.0) and Windows 95/98/NT/2000/XP [46].

A C++ compiler is needed to compile and run the Festival scripts and modules. The current compilers used for the different platforms discussed above are as follows [46]:

- Sun Sparc Solaris distributions: GCC 2.7.2, GCC 2.8.1, SunCC 4.1, egcs 1.1.1, egcs 1.1.2
- Sun Sparc SunOS 4.1.3: GCC2.7.2
- Intel SunOS 2.5.1: 2.7.2
- FreeBSD for Intel 2.2.1, 2.2.6 and 3.x (ELF based): GCC 2.7.2.1
- Linux 2.0.3 for Intel (RedHat 4.1, 5.0, 5.1, 5.2, 6.0): GCC 2.7.2, GCC 2.7.2, egcs 1.1.1, egcs 1.1.2
- Windows NT 4.0 and 95/98/2000/XP: GCC 2.7.2 plus egcs, from Cygnus GNU win32 b19, Visual C++ PRO v5.0

The recommended compiler is GCC 2.7.2 since it compiles faster and produces better code than previously tested compilers [46]. The GNU *make* program is the preferred *make* program since it has been tested on all the systems. It is also stated by the authors of [46] that they prefer this *make* program because it has been trusted to work well.

Audio hardware support systems are in abundant for Festival. These are inherited from the audio support available with the Edinburgh speech tools library. The current supported methods are:

- NCD's *NAS*, a network transparent audio system.
- */dev/audio*, a simple low level method for audio output that is limited to *mu-law* encoding at 8KHz. Offered by Linux and BSD machines.
- */dev/audio (16bit)*, 16 bit linear audio at different sample rates. Supported by Sun Microsystems.
- */dev/dsp (voxware)*, supported by FreeBSD and Linux machines. Has a compile support in the Edinburgh speech tools library that must be compiled before Festival can make use of it. Can be set to 16 bit linear audio for Linux machines.
- *mplayer*, supported Windows machines. If *mplayer* is not available then the Windows machine will use the default audio system *win32audio.*

Software requirements

The software source packages required to run and build new TTS systems using the Festival speech synthesis system are:

- *Festival-1.4.3.tar.gz:* The Festival speech synthesis system source files.
- *Speech-tools-1.2.3.tar.gz:* The Edinburgh Speech Tools library.
- *Festlex_NAME.tar.gz:* Lexicon distributions of different voices available in Festival.
- *Festvox_NAME.tar.gz:* The speech databases of the different voices available in Festival.
- *Festvox2.0.tagzr:* Script files and modules needed for building new TTS systems.
- *Festdoc_1.4.3.tar.gz:* Full system documentation for Festival and the EST.

Each package is freely available for download from [9] and [52]. Now that we know how the Festival speech synthesis system operates, how its architecture is structured and what its requirement are, we can look at the techniques used to implement concatenative speech synthesis. Both DCS and USS synthesis techniques are discussed in detail in the following section, and the idea of a hybrid TTS system is also discussed.

3.2 Text-To-Speech synthesis techniques in Festival

The Festival speech synthesis system has two techniques of implementing concatenative speech synthesis. The first method is called diphone concatenative synthesis and simply uses the diphones of a language as concatenation units for synthesis. The second method called unit selection synthesis allows the system designer to use different unit sizes as the units for concatenation and synthesis. The unit sizes vary from phone to diphones, syllables and even full words. The units are obtained by recording natural, fluent speech into a speech database. Both these techniques are discussed in detail in this section and a discussion is also made on the idea of a hybrid system that combines the advantages of the two techniques.

3.2.1 Diphone concatenative synthesis (DCS)

By definition the diphone is the phone to phone transition from the middle of the first phone to the middle of the second phone [7]. This means that phones are joined at their stable steady state regions and are therefore easy to join [11] [53]. The amounts of diphones present in a language are the square of the number of phones present in the language [6] [7]. Festival uses carrier words also known as non-sense words to house the diphones to be used for synthesis. Since there is only one example of each diphone available for synthesis, Festival uses target word embedded carrier sentences (the non-sense words) to guarantee that the diphones are recorded with a suitable duration and prosody. An example of the non-sense word "*ababa*" which contains the diphones transitions "*a-b*" and "*b-a*" is shown in Figure 3.3. This is a waveform representation of the recorded non-sense word using *wavesurfer*[10]. The full list of diphones used for the Afrikaans language is based on the phonetic descriptions discussed in Chapter 1 and the construction of the diphone database to be used is discussed in Chapter 4.

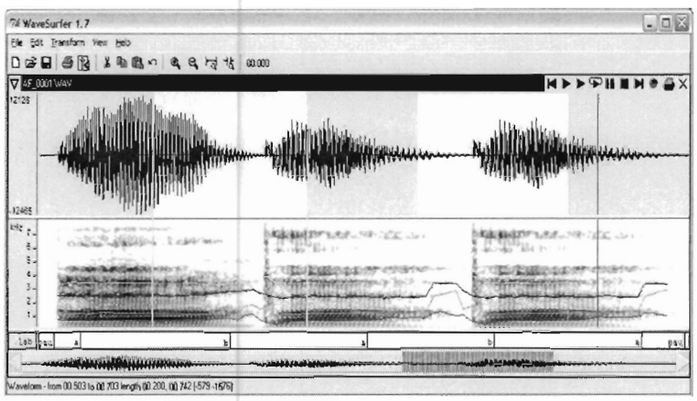


Figure 3.3: Waveform representations of nonsense word "*ababa*"

The Festival speech synthesis system uses Residual Excited Linear Predictive Coding (residual LPC) for the re-synthesis of diphones [46]. This technique assumes that a current speech sample x*(n)* can be predicted from a finite set of previous *p* samples,

[10] wavesurfer – an Open Source tool for sound visualization and manipulation [54]

x(n-1) to *x(n-k)*, by a linear combination with an error term *e(n)*, which is known to be the residual signal [3] [6]. This prediction is then formulated as:

$$x(n) = e(n) + \sum_{k=1}^{p} a(k)x(n-k) \tag{3.3}$$

and

$$e(n) = x(n) - \sum_{k=1}^{p} a(k)x(n-k)$$

$$= x(n) - \tilde{x}(n) \tag{3.4}$$

where $\tilde{x}(n)$ is the predicted value, p is the linear predictor order and $a(k)$ are the linear prediction coefficients (found by minimizing the sum of the squared errors over the speech frame) [3] [6].

The technique of diphone concatenative speech synthesis has been the dominant research technique of concatenative speech synthesis for many years [10]. The reason for this is because of the high flexibility and intelligibility that the technique offers. Systems built on the basis of this technique have the potential to synthesize or attempt to synthesize any linguistic entry and this is its greatest advantage. Diphones are relatively easy to join [11], and the memory requirements needed for the speech database (diphone database) is not as high as for other techniques since only one example of each unit is needed [7]. The technique has however become outdated due to the lack of naturalness and understandability in the resulting speech quality that it produces [10]. These two disadvantages are mainly contributed to the fact that only a limited amount of speech data is available for synthesis, and due to the expensive signal processing requirements needed for the determining prosody for the desired output [10]. Unit selection speech synthesis has taken over from diphone synthesis since the technique can provide synthetic speech at a much higher voice quality than diphone synthesis [12] [13] [14]. Recently more research and development has gone into USS. The two different forms of USS are discussed in detail in the following sections.

3.2.2 Limited domain unit selection synthesis (Ldom USS)

This is the first of two unit selection techniques employed by the Festival speech synthesis system. Limited domain USS has the advantage that it can produce the high quality synthetic speech that is required for real world speech applications [5] [13]. However, even though this technique produces TTS system with the highest voice quality it can still not produce all the qualities that are needed for advanced TTS systems since it can not produce the flexibility that is required [5]. The reason for this is that the technique can only synthesize words in a limited vocabulary (restricted database). This is the main disadvantage of this technique and can therefore not be used to build general TTS system for multiple applications [5] [13].

Instead of using smaller units such as phones or diphones this technique uses the unit type *phone_word* (phone plus word) and not full words as it may seem [42]. This technique identifies a phone depending on the word it comes from by looking at the position of each phone within the word to be synthesized. If a phone and its position can not be identified with any word in the database then the word is labeled as an out of vocabulary word and can therefore not be synthesized [7]. In Chapter 2 different unit selection algorithms are discussed. The Festival speech synthesis system makes use of a unit selection algorithm called *cluster unit selection* to select the appropriate unit for synthesis. This algorithm builds clusters of similar units by grouping them based on their acoustic similarities [7]. The grouping is based on the phonetic context, prosodic features (F0 and duration), higher level features such a stressing, utterance positions and accents [7]. The technique works similarly to that of the CHATR unit selection algorithm described in Chapter 2. This technique however avoids the calculation of the target costs by pre-building CART trees to select the appropriate cluster candidate of units for synthesis. Since these clusters are built directly from acoustic scores and target features, the target feature estimation function is not required hence there is no need for the calculation of the feature weights of each feature. Therefore, the advantage of this technique is that it selects a group of candidates, and finds the best unit by finding the best path through each set of candidates for the target unit to be used for synthesis. This results in the technique working faster and more effectively than the CHATR technique,

and it also produces synthesis at a smaller time delay time. The technique also uses a spectral smoothing technique called "*Optimal Coupling*" that helps in unit selecting process by find the most appropriate acoustic join of two units at run time [7] [38] [55].

One limitation in this technique is contributed to the method used to handle out of vocabulary words. The system can either use a backup voice (usually a diphone voice) or a back phrase to handle out of vocabulary words. If neither of these is in place the system will fail. The use of a back-up voice has lead to the realization of a hybrid system that attempts to synthesize any linguistic entry to the system. The use of such a system is fully realizable within the Festival framework and has been implemented by [5]. This technique is discussed later in this chapter.

3.2.3 Open domain unit selection synthesis (Odom USS)

This is the second technique of unit selection synthesis available with the Festival speech synthesis system. The technique has the ability to produce TTS systems with a high degree of understandability, naturalness and pleasantness, but has the disadvantage that it can be inconsistent [10]. The technique can also be seen as an upgrade of the DCS technique. The speech quality that the technique can produce does however depend on the quality of the speech database units are selected from [56]. As with all TTS systems the final voice quality of the system is highly dependant on the speech database used [14]. The authors of [12] states, "*One approach to the generation of natural-sounding synthesized speech waveforms is to select and concatenate units from a large database*". Therefore using a well defined speech database, that has a good phonetic coverage, can produce high quality TTS systems.

The technique uses the same *cluster unit selection* strategy used for the limited domain technique. The difference here is that the size of the speech units used for synthesis is variable from phones to diphones and even syllables. Instead of using full words (*phone_word* as described earlier) as speech units, this technique uses the sub units of words. Acoustically similar units are clustered into groups, and the target unit is selected

by finding the best path through each set of candidates. It is also possible to use the same speech database used for an Ldom system, provided it has a good phonetic coverage.

Recently more work has been done on creating optimal databases for systems based on this technique, and algorithms for predicting intonation and accents have been developed to improve the overall qualities of open domain unit selection speech synthesis systems [41] [51].

3.2.4 Hybrid Text-To-Speech Synthesis

This form of speech synthesis is not a direct technique available with the Festival speech synthesis system. It is defined here and throughout the rest of this thesis as means of eliminating the disadvantage of the lack flexibility in limited domain systems by using a back-up system for synthesizing out of vocabulary words. This kind of system provided a complementary back-up system to the Ldom system can be found, has a great potential of meeting all the requirements needed for an Advanced TTS system.

The Festival framework allows for the design of a hybrid system of the Ldom and DCS systems. This system uses a DCS system as a back-up system to an Ldom system. Figure 3.4 shows how the advantages of these two techniques can be added to meet all the requirements of an advanced TTS system. The resulting system will have the flexibility and intelligibility of a DCS system accompanied with the understandability, naturalness and pleasantness of an Ldom unit selection synthesis system.

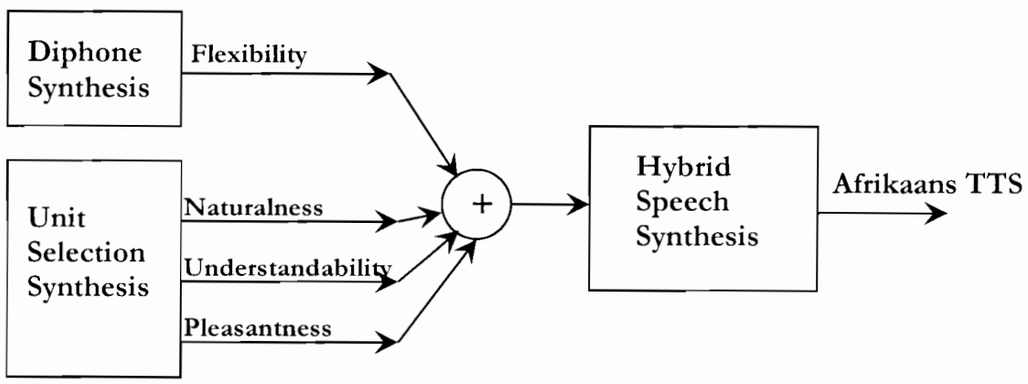


Figure 3.4: Hybrid Ldom/DCS speech synthesis system [5]

A possible downfall of this technique is the quality of the back-up diphone system. The reason for this is because the DCS system will not have the naturalness, pleasantness and understandablity to complement the Ldom system. One possible solution to this problem would be to use an open domain unit selection system as the back-up system to the Ldom system. This technique can produce the naturalness, pleasantness and the understandability that is needed to complement the Ldom system also with a high degree of flexibility. Such a system has been implemented by [57], and is used within their *SmartKom* dialog system. Figure 3.5 shows the combination of the advantages of the two techniques into one hybrid system that meets all the requirements of an advanced TTS system.

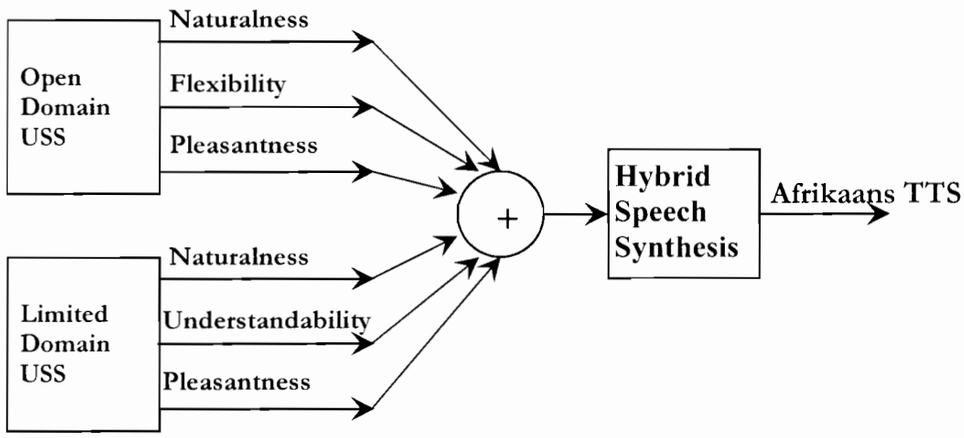


Figure 3.5: Hybrid Ldom/Odom speech synthesis system

3.3 Summary

This chapter introduces and discusses the Festival speech synthesis system. The hardware and software requirements are discussed in detail in section one. Section two discussed the different concatenative synthesis technique available with the system and what each technique's advantages and disadvantages are. Each of the techniques discussed are implemented in the design of an advanced TTS system for Afrikaans, and the two hybrid approaches discussed last are also implemented in the same aim. These implementations and the various systems designed are discussed in Chapter 4.

CHAPTER 4

Implementation of Afrikaans TTS systems

The Festival speech synthesis system and its techniques of concatenative speech synthesis provide the framework needed for building advanced TTS systems in different languages. It was therefore used for the design and implementation of the advanced TTS system for Afrikaans. The techniques discussed in Chapter 3 were each implemented in an attempt to build one system that meets all the requirements needed for an Advanced TTS system. This chapter discusses the steps involved in designing each Afrikaans TTS systems, and do note that each system was built in a laboratory environment and no professional equipment was used. As described in Chapter 3 the following voice building tools and speech analysis tools are required for building new TTS system in Festival.

1. Festival-1.4.3.tar.gz
2. Speech-tools-1.2.3.tar.gz
3. Festvox2.0.tag.zr

Each of these packages is freely available for download from [9] and [52]

4.1 The Diphone Concatenative Afrikaans TTS system

This system uses the diphones of the Afrikaans language as concatenation units for synthesis. The system is defined by the three identifiers; institution name, intended language, initials of the system designer and the synthesis technique used. For short these identifiers are abbreviated to *INTS_LANG_INITIALS_SYNTECH*. The Afrikaans DCS is

therefore defined as *uct_af_fer_diphone*[11] (see Appendix A.1 for system definition directories).

4.1.1 Constructing the Diphone database

This step involves the definition of the diphone database consisting of all the possible phone-to-phone transitions for the Afrikaans language. Since diphones are the simply the phone-to-phone transitions the first step was to define the phoneset for the language [7]. The phonetic transcriptions described and discussed in Chapter1, Section 1.2.3 were used as the phoneset for this task. The non-sense words used for housing the diphones are then created by the diphone generation schema file *af_schema.scm* (see Appendix A.1 */uct_af_fer_diphone/festvox/af_schema.scm*) using the consonant-to-consonant (CC), consonant-to-vowel (CV), vowel-to-vowel (VV) and vowel-to-consonant (VC) transition rules for the language. These transition rules were found with the use of the Afrikaans dictionary "*Groot Woordeboek*" [6] [58]. An example of a non-sense word for Afrikaans containing the diphone transitions "*a-b*" and "*b-a*" and silence phones *pau* is shown in Figure 4.2 below. The full list of non-sense words are shown in Appendix A.1 */uct_af_fer_diphone/etc/afdiph.list.*

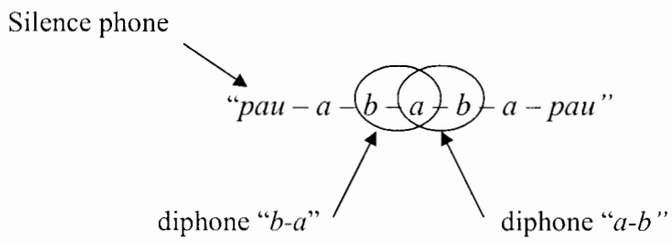


Figure 4.1: Non-sense word "*a-b-a-b-a*"

4.1.2 Recording the speaker

This step involves the acquisition of the speech database for synthesis. In this case we record the non-sense words generated in the previous step. This is a very important step in the voice building process because the final voice quality of the system is dependant

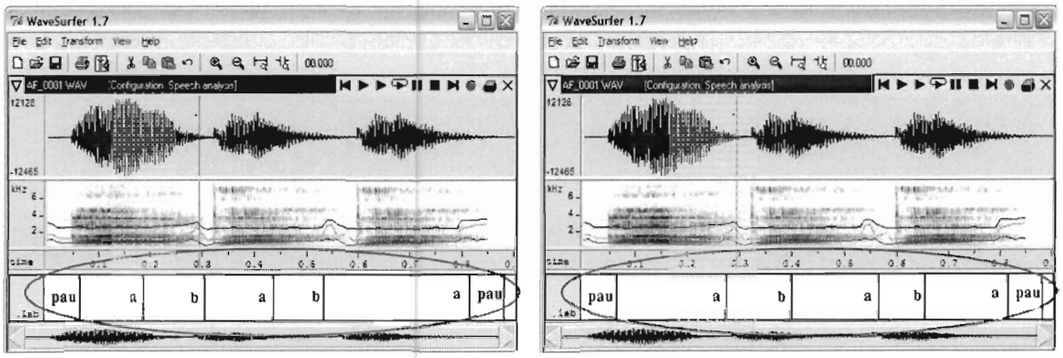
[11] *uct_af_fer_diphone – university of cape town_afrikaans_francois enrique rousseau_diphone synthesis*

on the quality of the speech database [14]. Using a professional recording studio with professional recording equipment and a professional speaker will produce the most desired speech database but can be very expensive. For this research we did not have access to the requirements mentioned above and therefore, we recorded straight to a standard Pentium 4, 2.8GHz machine at a sample rate of 16 000 samples per second, with a standard microphone and the voice talent of the author of this thesis in a laboratory environment. The *na_record* recording tool, part of the EST library [48] was used to record the database. This recording tool creates a log of all the recorded data and stores them into wave format audio files (see Appendix A.1 */uct_af_fer_diphone/wav*).

4.1.3 Labelling the speech database

The authors of [59] state the following, "*The quality of a concetenative text-to-speech system is directly related to the accuracy with which the underlying acoustic inventory is labeled*". The objective of the labeling process is to align the speech unit boundaries with the recorded database so that the system will know where in the database the units for synthesis must be extracted from. In the case of DCS this process marks the positions of the diphone boundaries for each diphone within the non-sense words [23]. Since these boundaries give the system information regarding the position of a diphone within a non-sense word, it is easy to understand the statement made by the authors of [59].

If theses boundaries are incorrectly placed then it would mean that the wrong regions of speech would be represented by these labels. The end result of this is that the incorrect regions of speech is be used at synthesis, and hence produce undesired results. This becomes a big problem for large databases since the hand correcting of labelling errors (fixing all the labelling errors by hand) is a very time consuming task and is also inconsistent [59] [60]. An example of an incorrectly labelled diphone with a corrected version to its right is shown in Figure 4.2.



(i) Incorrectly labelled diphone (ii) Correctly version

Figure 4.2: Corrected labelling of incorrectly labeled non-sense word "*ababa*"

The incorrectly labelled diphone shown above-left produces undesired results since the incorrect regions of speech are represented by the labels. Correcting these labels ensures that the appropriate regions of speech are used for synthesis, and hence produces more desired results.

The Festival speech synthesis system provides two automatic speech labelling algorithms to perform the task of labelling speech databases. The first algorithm uses the technique of Dynamic Time Warping (DTW)[12] to find the phone boundaries of a recorded utterance by aligning it with a previously synthesized utterance where phone boundaries are already known [7]. This is done by first converting each utterance into its acoustic feature representations. Here the acoustic features used are the Mel Frequency Cepstral Coefficients (MFCCs) and the delta MFFCs (the difference between a current MFCC vector and the previous MFCC vector). A euclidean distance measure is then used to define the distance between the feature vectors of the two utterances. The DTW algorithm is then used to align the feature vectors of the two utterances. Once the two utterances are time aligned the diphone boundaries of the synthetic utterance is mapped to the recorded utterance [7] [61].

[12] DTW – a time aligning technique of two speech signals [7] [61]

The second technique uses Hidden Markov Model (HMM)[13] training using Sphinx[14] and SphinxTrain[15] [62] [63]. This technique uses Automatic Speech Recognition (ASR) tools to build acoustic models of the speech data in the database. This is done by using a known sequence of phoneme models to generate new phonetic alignments for the speech database [7]. It is further discussed in Section 4.4.

It is said that DTW works best for smaller databases and is computationally less expensive than HMMs while the latter work well for databases in different languages [7] [64]. The DTW technique was used for the design of the diphone concatenative Afrikaans TTS system since the database needed for this technique is much smaller compared to the databases for unit selection systems. Each labelled diphone was then inspected, and incorrect labels were hand corrected using *wavesurfer* to ensure that the correct coverage of the diphone is used for synthesis. See Appendix A.1 */uct_af_fer_diphone/lab* for labelled database.

4.1.4 Building the diphone index

The diphone index contains a time index for each of the diphones in the database which based on the labels gives the system information on where in the non-sense word the diphone should be extracted from [7]. By default the diphone is extracted from the middle of the first phone to the middle of the second phone. The reason for this is that only the co-articulate regions of the phones (stable regions) and the transition are to be used for synthesis [6] [8]. In 2004 the author of this thesis published a paper on extending the region of the phones used for synthesis. The paper entitled, "*Increased Diphone Recognition for an Afrikaans TTS system*" was presented and the annual Patten Recognition Association of South Africa (PRASA) conference for 2004 (*Appendix E.1*). It argued the fact that instead of using the default regions for synthesis, the full phone to phone transition should be used. An advantage to be gained by this approach is that all the information regarding the phone-to-phone transition is used for synthesis and not just a percentage thereof.

[13] HMM - acoustic modeling of speech using state sequences
[14] Sphinx – Speech Recognition System http://cmusphinx.sourceforge.net/sphinx2/
[15] SphinxTrain – Sphinx training pakage [62]

The extension of the diphone coverage was achieved by modifying the script provided by Festival for building the index. To calculate the diphone boundary (DB) shown in Figure 4.3 the script *make_diph_index* (see Appendix A.1 */uct_af_fer_diphone/bin/make_diph_index*) uses the equation:

$$DB = (y+z)/2.0 \tag{4.1}$$

This equation was then changed to:

$$\text{Let } DB = z, \text{ instead of (4.1)} \tag{4.2}$$

This ensured that the entire diphone is now used for synthesis and not just the portion shown in Figure 4.2.

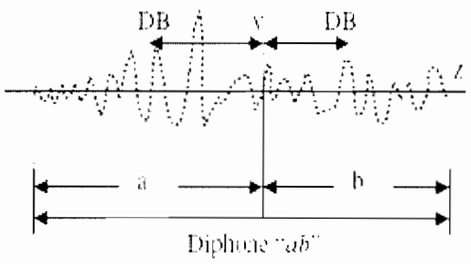


Figure 4.3: Diphone boundaries for the diphone "*a-b*" [6]

The disadvantage of this approach was realised when a comment was made by a member of the audience at PRASA. The comment was that all phones would now be pronounced twice. This comment was entirely credible since the concatenation of a diphone with another of the same kind would result in the transition to be pronounced twice. Example, the word "*ababa*" would now be pronounced "*ab-ba-ab-ba*" instead of "*ababa*" as intended. It was therefore decided to revert back to the original definition of the diphone index. The diphone index for the diphones, "*ab*" and "*ba*" are shown below. The index (in seconds) shows the diphone name, the file identity, the start time (mid point of the first phone), the phone boundary (transition at the centre of the diphone), and the end time (mid point of the second phone). For full index see Appendix A.1 */uct_af_fer_diphone/dic/afdiph.est*

b-a af_0050 0.737924 1.0113 1.09832
a-b af_0050 1.09832 1.18534 1.38061

4.1.5 Extracting the Pitchmarks

As described in Chapter 3, Section 3.2.1 diphone concatenative synthesis uses Residual Excited Linear-Predictive Coding for the re-synthesis of diphones. This technique is a pitch synchronous technique that requires information on the positions of the pitch periods (pitchmarks) present in an acoustic signal (recorded non-sense word in this case) [7]. The EST library supports two scripts for pitchmark extraction from waveforms. The first script called *make_pm* extracts pitchmarks from an electroglottograph[16] (EGG) recorded signal. This signal contains the electrical activity in the glottis which eases the process of finding the pitchmarks and produces more accurate results [7] [53]. The second script is a modified version of *make_pm* (has higher order filters) that extracts pitchmarks from a raw acoustic signal where an electroglottograph was not used. This technique produces less accurate results than when an EGG is used but is still good enough to be used for the task at hand [7].

For this research no electroglottograph (laryngograph) was available and therefore the pitchmarks were extracted from the raw acoustic signals (recorded non-sense words) using the modified version of the *make_pm* script, *make_pm_wave*, provided by the EST. Bad pitchmarks extraction can result in bad synthesis and therefore the pitchmark fixing script *make_pm_fix* was used to fix bad pitchmarks move them to the nearest peaks [7]. See Appendix A.1 */uct_af_fer_diphone/bin* for these two scripts.

4.1.6 Building the LPC parameters

This step of the procedure in building the DCS system is needed for the extraction of the LPC parameters (LPC coefficients) and the LPC residuals from each of the non-sense words in the diphone database needed for the re-synthesis of diphones [7]. The LPC analysis is done pitch-synchronously which is the reason for extracting the pitchmarks in the previous step. An important step in the determination of the LPC coefficients and its residuals is the normalisation of the power factors in each of the non-sense words. Power fluctuations between non-sense words occur due to inconsistent recording by the speaker,

[16] EGG – Apparatus used to measure the electrical activity in the glottis during speech [7]

recording in a noisy environment and using bad recording equipment [5] [7]. The power normalisation technique used by Festival calculates the mean vowel power for each non-sense word database, and then uses this mean to calculate the average factor difference for each vowel in the database and finally scales the non-sense words according to this average factor.

After power normalization the LPC coefficients were obtained using the power normalised non-sense words and the EST signal-to-feature vector program *sig2fv* [48]. The residuals were then calculated using the EST program *sigfilter* which finds the residuals by inverse filtering the recorded non-sense words [8].

4.1.7 Building the lexicon support database

As described in Chapter 3, Section 3.1.2 the lexicon is a subsystem that provides pronunciation definitions for the words in a language. It consists of an addenda, a compiled lexicon and a set letter to sound rules. For the DCS Afrikaans TTS system the words given lexicon definition were the ones used in the test sentences that are used for testing the system. The reason for this is that building a large lexicon for an entire language can take up to weeks to complete. A set of letter to sound rules (LTS rules) for the language were also defined here. The lexicon entries for the first test sentence are shown below (the full lexicon and LTS rules can be seen in Appendix A.1 */uct_af_fer_diphone/festvox/uct_af_fer_lexicon.scm*):

```
(lex.add.entry '("Afrikaans" nn (((A f r i k a n s) 0))))
(lex.add.entry '("is" nn (((I s) 0))))
(lex.add.entry '("die" nn (((d i) 0))))
(lex.add.entry '("moedertong" nn (((m u d I r t O N) 0))))
(lex.add.entry '("vir" nn (((f I r) 0))))
(lex.add.entry '("veertien" nn (((f e r t i n) 0))))
(lex.add.entry '("persent" nn (((p I r s E n t) 0))))
(lex.add.entry '("van" nn (((v A n) 0))))
(lex.add.entry '("Suid" nn (((s & d) 0))))
(lex.add.entry '("Afrika" nn (((A f r i k A) 0))))
(lex.add.entry '("se" nn (((s I) 0))))
(lex.add.entry '("bevolking" nn (((b I f O l k I N) 0))))
```

These lexical entries now ensure that the first test sentence (and the rest) is pronounced according to phonetic transcriptions for the Afrikaans language.

4.1.8 Assigning duration and intonation

As mentioned in Chapter 3 the addition of intonational accents adds accent information to the syllables of an utterance. It was decided to use the default intonation module discussed here since accents stressing is added in the lexicon definitions. Duration information was added using the CART duration model also discussed in Chapter 3. The reason for using this model instead of a consistent default value is that more natural synthesis is achieved when using natural durations [65]. The CART model firstly calculates the mean and standard deviations of all the Afrikaans phones in the diphone database, and then applies equation 3.2 to calculate the average duration to be used at synthesis.

The final stage in building the DCS Afrikaans TTS system was to test the system using the testing procedure discussed in Chapter 5. The results and the evaluations of the results obtained for this system are also discussed in this chapter.

4.2 The Limited Domain Unit Selection Afrikaans TTS system

This was the second system to be implemented. Originally this system was built to experiment with the high voice quality promised by [5] [13] [57], but it was later realised that more could be achieved with the technique than just building a restricted domain TTS systems. Discussed in this section are the steps involved in the design of a limited domain (Ldom) unit selection synthesis for Afrikaans. The system is defined as *uct_time_af_ldom* (see Appendix A.2).

4.2.1 Designing the database

Since limited domain USS systems have restricted outputs the first step in building the Afrikaans system was to define the vocabulary for the desired output. The construction of the vocabulary database is discussed here and is in the form of a prompt list that covers

the desired output. It is stated by [13] that it is important to know what the desired application of the synthesiser is going be so that the prompt list can be defined to cover the desired output sufficiently. The prompt list designed for the Afrikaans Ldom system was an extension of a list designed for a demonstration of an Afrikaans TTS system to the South African State Information Technology Agency (SITA). The database consists of 400 sentences of which the first 24 were used for the SITA demonstrations. These 24 sentences included greetings, general information about SITA and the Afrikaans languages. The system is independently available in Appendix A.2 and is defined *uct_time_sita_ldom*. The full a database for the Afrikaans Ldom system is shown in Appendix A.2 */uct_time_afr_ldom/etc/time.data,* and was sourced from an online Afrikaans newspaper, *Die Rapport* [66] and information website Wikipidi [67].

The prompt file used to define the prompt list is called *time.data* and is used by the designers of Festival for building talking clocks [7]. This file was modified to hold the Afrikaans prompts by replacing each of the prompts in *time.data* with the prompts of the Afrikaans prompt list [5].

Examples of the first four prompts in this file are:

(time0001 "nul, een, twee, drie, vier, vyf, ses, sewe, agt, nege")
(time0002 "goeie, more, dames, en, minere")
(time0003 "jou, telefoon, nommer, is")
(time0004 "jou, identiteits, nommer, is")

Each prompt is given an identity with the term *time000**. This identity is used to label the prompt so that it can be identified by the sub-processes.

4.2.2 Synthesizing the prompts

This step involves the construction of the synthetic prompts needed for recording the vocabulary from the prompt list, and also needed to label the resulting speech database. The *kal_diphone*[17] American diphone system was originally used to generate the synthetic prompt for each prompt in the database. This gives the system designer a

[17] An American diphone voice developed by the designers of Festival

reference pronunciation for each prompt, a time limit for recording each prompt and reference labels for labelling the resulting database. The synthetic prompt generation system was later changed to the Afrikaans diphone system, *uct_af_fer_diphone,* because the phonetic transcriptions of the American diphone voice did not match the phonetic transcriptions needed for the Afrikaans Ldom system.

4.2.3 Recording the speaker

The same voice talent, recording equipment and recording environment used to record the diphone database for the DCS system was used to record the database for the Ldom system. The reference pronunciations provided by the synthetic prompts were not used for the reason that it adds a redundant time factor that is not needed if the voice talent is familiar with language. Each prompt in the prompt list was recorded on a word by word basis. This was achieved by placing a comma (see extract from prompt list above) between words which ensures that a silence in placed between consecutive words. The reason for using this strategy was to ensure that overlapping of consecutive words does not occur as when recording in a fluent manner. The overlapping of words makes the task of labelling the speech database more complicated since it is not known where a singular word starts and ends. As discussed earlier, incorrect labelling results in undesired regions of speech being used at synthesis. Usually limited domain USS systems use natural fluent recorded databases that can be used for specific applications [57], but since this was not the aim of this project this strategy could not by applied.

4.2.4 Labelling the speech database

The Dynamic Time Warping (DTW) speech labelling algorithm was used to label the speech database for the Ldom Afrikaans TTS system. As in the case of the DCS system, both the synthetic prompt and the recorded utterance were first converted into their respective MFCC's and delta MFCC's acoustic features, and then time aligned by the DTW algorithm [61]. The utterances are each labelled on a phonetic basis with each word represented by the phones that make up the word. The reason for this is that the Ldom technique uses the unit type *phone_word* (phone plus word) as described in Chapter 3.

This strategy needs information on the position of a phone within a word for synthesis and it is for this reason that all the phones in each utterance must be labelled [7].

4.2.5 Generating acoustic features

The cluster unit selection algorithm discussed in Chapter 3 requires information on the acoustic features of the utterances in the database to groups units based on their acoustic similarities. The acoustic features of speech used here are the mel-scale cepstral coefficients and are extracted pitch synchronously rather than at fixed rates. The reason for this is because it was discovered that the unit selection technique produces better results if parametric spectral representation are represented at pitch periods [7] [13]. Therefore, the pitchmarks in the recorded utterances must be extracted. No EGG was available as before and therefore the EST *pitchmark* extraction script *make_pm_wave* was used to extract the pitchmarks from the raw utterances. The bad picthmarks were then fixed using the same *make_pm_fix* script used for the DCS system. Power normalization was then performed on each of the utterances in the database to ensure that there is no power mismatch between words of different utterances. Finally the mcep parameters were extracted using the *make_mcep* script provided by Festival.

4.2.6 Building the unit clusters

This section discusses the steps used by the automatic clustering algorithm to automatically build clusters of similar units (as described in Section 3.2.1). The first step loads all the utterances in the database, sorts each utterance into unit type (*phone_word* in this case) and then assigns an individual name to each unit so that they can be identified by the sub-processes [7]. The second step loads the acoustic parameters (MFCC's plus F0), calculates the acoustic distance (Euclidean Mahalanobis Distance [limited, building]) between units of the same type, and saves these distances into a distance table. The third step builds cluster indexes to clusters using the feature descriptions (high-level features such as phonetic and prosodic features [68]) of each unit. The fourth step finds which features best minimizes the acoustic distances between units by looking at the relationship between the features and the distances. This is done with the use of the

CART tree builder program called, *wagon*. This program builds a decision tree that minimizes the acoustic distance between units of the same type [7] [13] [69]. The final step is to take all the trees generated in step four, place it into a single file to be added to a unit catalogue that consists of a list of all unit names, the utterance they come from and the position within in the utterance. All this is carried out by running the Festival script *build_clunits* described in [7].

The final stage in building the limited domain unit selection synthesis system for Afrikaans was to test the system using the testing procedure discussed in Chapter5.

4.3 The Hybrid Ldom/DCS system

The motivation behind this approach was that the resulting system would have all the qualities needed for an advanced TTS system for Afrikaans. This system combines all the advantages of the Afrikaans Ldom system with the advantages of the Afrikaans DCS system, and is shown in Figure 3.4. This section discusses the objectives of the hybrid system and the steps involved in building the hybrid Ldom/Odom Afrikaans TTS system using the techniques within the Festival framework.

4.3.1 Objectives of Hybrid TTS system for Afrikaans

The objectives of the hybrid system as discussed by [5] are:

1. The system must be in the domain of the limited domain system at all times.
2. The system must show its flexibility by using the DCS system as a back-up system to synthesize out of vocabulary words, not in the limited domain.
3. The system must revert back to the limited domain system after an utterance is successfully synthesized.

These objectives are illustrated in the flow diagram below. This is followed by a discussion on the basic operation of the system.

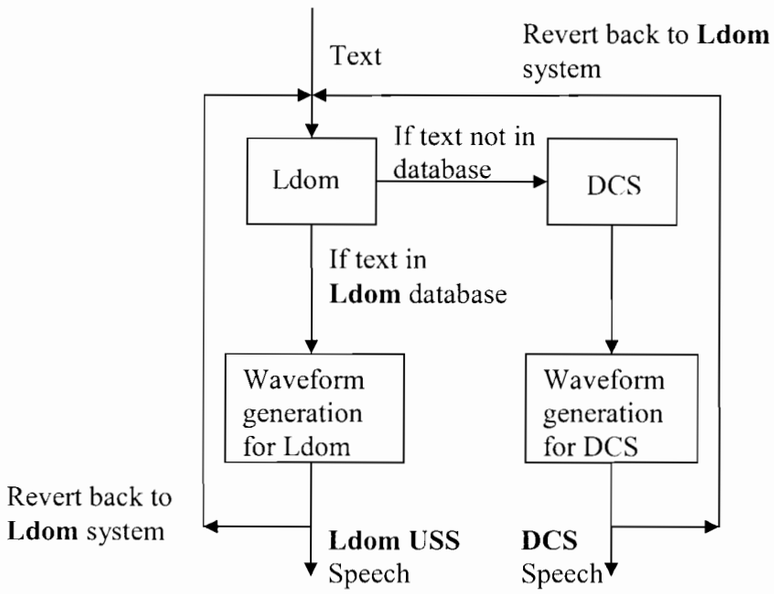


Figure 4.4: Objectives of Hybrid TTS system [5]

When the text to be synthesized is entered, the front end of the Ldom system acts as the decision block that decides which system will be used to synthesize the text. If the text entered is not in the database of the Ldom system then it will synthesized by the DCS system. This meets the second objective of the hybrid system. However, if the text entered is in the Ldom database, then the word will be synthesized by the Ldom system. After synthesis is performed by either system it reverts back to the Ldom system, until the next utterance is ready to be synthesized.

4.3.2 Building the hybrid Ldom/DCS Afrikaans TTS system

The steps involved in designing the hybrid system are exactly the same as the steps involved in designing the Ldom TTS system discussed in Section 4.2. The Ldom built in Section 4.2 automatically becomes a hybrid system when changing the synthetic prompt generation system as discussed, but the system does however not meet all the objectives that are discussed here. The problem is that the system does not revert back to the Ldom system after DCS synthesis and can therefore not be used for the task at hand. This

section therefore discusses the extension of the system built in Section 4.2 to the full hybrid Afrikaans TTS system that meets all the objectives discussed.

The first step in this process was to change the definition of the synthetic prompt generation system used to act as the back-up system to the Ldom system. This is done by modifying the festival script *uct_time_afr_ldom.scm* (see Appendix A.2 */uct_time_afr_ldom/festvox/uct_time_afr_ldom.scm*) used for the construction of limited domain systems. The line of code in the script, used to define the back-up system was changed to use the DCS Afrikaans system, *uct_af_fer_diphone*. This is shown below (an extract from the script, *uct_time_afr_ldom.scm*)

(set! uct_time_afr::closest_voice 'voice_uct_af_fer_diphone)

By changing the back-up system, we meet the second objective of a hybrid system. The third and final objective was met by modifying the same script to ensure that the system reverts back to the Ldom voice after synthesis. The original set up of the script makes the system come to a complete halt after synthesising an out of vocabulary word with the DCS system. The problem lies in lines 5 of code extracted from the script, *uct_time_afr_ldom.scm* shown below.

```
(define (voice_uct_time_afr_ldom)
 "(voice_uct_time_afr_ldom)
Define voice for limited domain: time."
```

line 4… ***(eval (list uct_time_afr::closest_voice))***

line 5… ***(uct_time_afr::domain_specific_voice_setup)***

```
(if (not (equal? utt.synth uct_time_afr::utt.synth))
 (begin
(set! tts_hooks (delq utt.synth tts_hooks))
(set! tts_hooks (cons uct_time_afr::utt.synth tts_hooks))
(set! uct_time_afr::real_utt.synth utt.synth)
(set! utt.synth uct_time_afr::utt.synth)))
```

Line 4 evaluates the system (Ldom or DCS) that the system is currently in. If false, meaning the system is in the limited domain voice then it will stay there. However, if

true, meaning the system is in the closest voice (*uct_af_fer_diphone*) it should then revert back to the Ldom system after synthesis. Instead, line 5 sends the system into the domain specific voice setup function which brings the system to a halt. To eliminate this problem this line of code was deleted, which ensures that the system switches between the limited domain and the diphone system as required by the objectives.

The paper entitled, "*A Hybrid Text-To-Speech System for Afrikaans*" was written on the resulting system and can be seen in *Appendix E*.2. This system was not tested using the procedure in Chapter 5. Only the comments made by the listening subjects were used for evaluations.

4.4 The Open Domain Unit Selection Afrikaans TTS system

To experiment with the two labelling algorithms discussed in Section 4.1.3, two different Odom Afrikaans TTS systems were built, one for each labelling algorithm. The systems were conveniently called *uct_af_fer_clunits* (DTW labelled system) and *uct_af_pau_clunits* (SphinxTrain labelled system). Described in this section are the steps involved in designing the two open domain unit selection TTS systems for Afrikaans.

4.4.1 Constructing, recording and labelling the speech database

The same 400 sentence phonetically balanced database used for the design of the Ldom Afrikaans TTS system, was used as the databases for the design of the two Odom systems. For this reason there was no need for re-recording the database for either system. See Appendix A.3, *uct_af_fer_clunits/wav,* and A.4, *uct_af_pau_clunits/wav.*

The database for the DTW-labelled system was re-labelled for the reason that we wanted independent labels between the systems implemented. The DTW speech labelling algorithm discussed in Section 4.1.3 was again used, and the synthetic prompt generation system was changed to the DCS Afrikaans system so that the phonetic transcriptions can match that of the Afrikaans phonetic transcriptions defined in Chapter 1.

The second Odom system, *uct_af_pau_clunits* was labelled using Sphinx and SphinxTrain which trains full acoustic HMM models on the database, and builds a data specific speech recognition engine to label the database [7]. The system was designed by the Carnegie Mellon University (CMU). The version of Sphinx and SphinxTrain used here are Sphinx2-0.4 and SphinxTrain-0.9.1 available from the CMU [62]. The first stage in this labelling process was to convert the speech database into a suitable format that Sphinx can understand. This is done by using *sphinxtrain* (Appendix A.4 */uct_af_pau_clunits/bin/sphinxtrain*), a program used by Sphinx to build sphinx files, convert waveform format, to do the actual training and to do the alignment and conversion of Sphinx labels back into the format that FestVox understands. The first step is to convert the Festvox MFCC's into the MFCC's format that Sphinx understands. SphinxTrain only supports raw header files such as NIST header files [7] and therefore, the MFCC's were converted to NIST header files. The next stage does the actual training, which consists of the following modules. For full explanation on each module see "*Sphinx-Instruction for training*" [63]

- Module 0, checks the basic files for training
- Module 1, builds vector quantisation parameters
- Module 2, builds context independent HMM phone modules by running the Baum-Welsh algorithm over the database
- Module 3, builds untied module definitions
- Module4, builds context dependant modules
- Module5a, build trees to question tied states
- Module5b, build trees for each state in each HMM
- Module6, prunes trees
- Module7, retrains the context dependant models with the tied states
- Module8, deleted interpolation

The next stage in the labelling process aligns the labels against the labels generated by the prompt stage. The final stage then converts the format of the Sphinx labels into the format that FestVox understands.

4.4.3 Building the utterance structures

The utterance structure is at the heart of the Festival speech synthesis system as discussed in Chapter3, Section 3.1.2. This step involves the construction of the utterance structures for utterances in the database. Required for the cluster unit selection system are the labels for the segments, durations and F0 targets. In this case the segments used were phones and it is therefore required that the phone labels are the same as the phones present in the phoneset for the Afrikaans language.

4.4.4 Generating the acoustic features

This step involves the extraction of the acoustic features from each utterance in the database needed for grouping similar units by the cluster unit selection algorithm. The mel-scale cepstral coefficients, *mcep* coefficients, are again used as in the case of the Ldom system and are again extracted pitch synchronously rather than at fixed rates for the reason that it produces better results as discussed earlier. The EST pitchmark extraction script, *make_pm_wave,* was again used and bad picthmarks were fixed using the *make_pm_fix* script. Power normalization was then performed to ensure that there is no power mismatch between utterances. Finally the *mcep* parameters were extracted using the same *make_mcep* (see Appendix A.3 */uct_af_fer_clunits/bin/make_mcep*) script used for the Ldom system.

4.4.5 Building the cluster units (clunits) from the utterances

The techniques and steps involved in building the clusters of each unit type for the two open domain system are exactly the same as the techniques used for the limited domain system. The difference here is that there are more units in the database and therefore takes longer than for the limited domain system.

Firstly the utterances are sorted into its phones and given individual group names. Then the acoustic parameters are loaded followed by the calculation of the acoustic distances between phones and are then placed in a distance table. The cluster indexes are then built using the feature descriptions of each phone. The relationship between the feature

descriptions and the acoustic distances between phones are then determined in the form of decision tress using *wagon*. The final step is to then list all the trees and to place them into a unit catalogue consisting of the phone names, the utterance it comes from and its position within the utterance.

The final step in building the two open domain unit selection TTS systems for Afrikaans was to test each system using the testing procedure discussed in Chapter 5. Comments are also made on which system performs best to give an indication of which labelling algorithm was more successful.

4.5 Extension of techniques into other South African languages

This section reports on advanced TTS systems for other South African languages using the same techniques used in building the advanced TTS system for Afrikaans. The systems were built by students employed by STAR[18] in the assist to build the first multilingual TTS system for all South African languages. Each student deserves full credit for implementing the techniques of Festival in each of their individual languages. The reports were written by each student, of which the author of this thesis had no influence on. The reports are shown in Appendix D.

4.6 Summary

This chapter discussed the implementation of each of the techniques of the Festival speech synthesis in the aim of designing an advanced TTS system for Afrikaans. One DCS system, one Ldom system, one hybrid Ldom/DCS system and two Odom systems were built. The details of the processes involved in building each system were discussed. The next chapter discusses the testing and evaluation procedure used to test and evaluate each of the systems implemented.

[18] STAR- Speech Technology And Research group, www.star.za.net

CHAPTER 5

Results and evaluation of each system

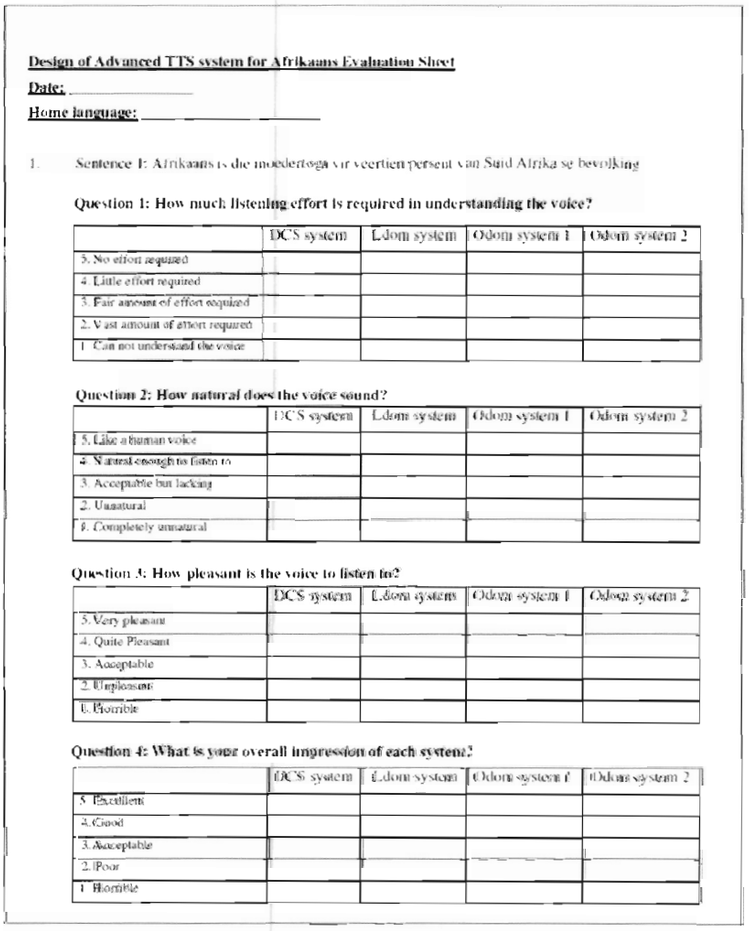
This chapter presents the results and evaluations of each system implemented in Chapter 4. Each system is tested and evaluated using subjective listening tests and six test sentences. Ten listening subjects were used to rate the understandability, naturalness and pleasantness of each system using a Mean Opinion Score (MOS) rating system. The flexibility of a TTS system can not be judged using subjective listening tests and therefore is only shown by the synthesis technique. The testing procedure used to test each system, the results and the evaluation of the results for each system are discussed in this section. The chapter is concluded with a proposal for a single technique and system that meets all the requirements of an advanced TTS system for Afrikaans.

5.1 Testing procedure

The overall objective of this thesis is to design an advanced TTS system for Afrikaans. Therefore, each of the techniques implemented in Chapter 4 must be tested for the requirements of an advanced TTS system for the language.

To perform the subjective listening tests six test sentences were constructed (available in Appendix B). Each sentence was played, using each system, to each of ten listening subjects (5 Afrikaans, 5 English/Afrikaans) individually. Each listener was then asked to complete an evaluation sheet based on a MOS rating system, to rate the quality of each system. The rating system used here is based on the modified MOS scale designed by

[70], used to measure the quality of TTS systems. Figure 5.1 shows page 1 of a six page evaluation sheet developed to evaluate the quality of each system. The first five pages of the evaluation sheet were used for the evaluation of the six sentences, and the final page was used to get the general comments of each system from each subject.



Design of Advanced TTS system for Afrikaans Evaluation Sheet

Date: _____

Home language: _____

1. Sentence 1: Afrikaans is die moedertong vir veertien persent van Suid Afrika se bevolking

Question 1: How much listening effort is required in understanding the voice?

	DCS system	Ldom system	Odom system 1	Odom system 2
5. No effort required				
4. Little effort required				
3. Fair amount of effort required				
2. Vast amount of effort required				
1. Can not understand the voice				

Question 2: How natural does the voice sound?

	DCS system	Ldom system	Odom system 1	Odom system 2
5. Like a human voice				
4. Natural enough to listen to				
3. Acceptable but lacking				
2. Unnatural				
1. Completely unnatural				

Question 3: How pleasant is the voice to listen to?

	DCS system	Ldom system	Odom system 1	Odom system 2
5. Very pleasant				
4. Quite Pleasant				
3. Acceptable				
2. Unpleasant				
1. Horrible				

Question 4: What is your overall impression of each system?

	DCS system	Ldom system	Odom system 1	Odom system 2
5. Excellent				
4. Good				
3. Acceptable				
2. Poor				
1. Horrible				

Figure 5.1: Page 1 of six age evaluation sheet

The ten test sentences used for evaluation are:

1. Afrikaans is die moedertong vir veertien persent van Suid Afrika se bevolking. (*Afrikaans is the home language to 14% of South Africa*)
2. Welkom by die demonstrasie van 'n Afrikaans sprekende rekenaar stelsel. (*Welcome to the demonstration of an Afrikaans TTS system*)

3. Die nege provinsies van Suid Afrika.
   (*The nine provinces of South Africa*)
4. Guateng, Noord Kaap, Oos Kaap, Wes Kaap, Kwazulu Natal, Limpopo, Noord Wes, Vrystaat, Mpumalanga.
   (*Names of the nine provinces*)
5. Kwazulu Natal het meer mense as enige ander provinsie in Suid Afrika.
   (*Kwazulu Natal has a higher population than any other province of South Africa*)
6. Dit is die einde van hierdie demonstrasie, dankie dat u geluister het, totsiens.
   (*This is the end of this demonstration, thank you for listening, good bye.*)

Systems were tested in the order that they were created. The testing procedure used is as follows:

- Step 1: Synthesize sentence 1 using the DCS system, and ask the listening subject to rate each question in the evaluation sheet by ticking the appropriate block. (*sent1DCS.wav*, Appendix B.1)
- Step 2: Synthesize sentence 1 using the Ldom system, and ask the listening subject to rate each question in the evaluation sheet by ticking the appropriate block. (*sent1Ldom.wav*, Appendix B.2)
- Step 3: Synthesize sentence 1 using the first Odom system (DTW-labelled), and ask the listening subject to rate each question in the evaluation sheet by ticking the appropriate block. (*sent1Odom1.wav*, Appendix B.3)
- Step 4: Synthesize sentence 1 using the second Odom system (HMM-labelled), and ask listening subject to rate each question in the evaluation sheet by ticking the appropriate block. (*sent1Odom2.wav*, Appendix B.4)
- Step 5: Tabulate the average results for all listening subjects for each system.
- Repeat steps 1-5 for each subject and the remaining five sentences.

Since the hybrid Afrikaans TTS system is a combination of the first two systems, and it will have the same results, it could not be tested in this manner. This system was therefore tested by asking each of the subjects whether they would be open to the use of the hybrid system.

5.2 Results and evaluation of results

This section shows the results of the subjective listening tests carried out to rate the performance of each system. Four tables are used to show the average result for the questions asked in the evaluation sheet. Therefore, one table each is used to show the average results of the understandability, the naturalness, the pleasantness and the overall impressions of each system implemented. The contents of each table are then showed in a graphical representation, which shows the comparisons between the systems. The yellow to orange transitions shadings in each graph represents the change in home language of the listening subjects used. Its moves from extreme Afrikaans (yellow - subject 1) to extreme English (orange – subject 10). The reason for this is to show how the two different language groups rates the quality of the systems. Evaluations on each system as per the requirements of an advanced TTS system are then made in the following section.

5.2.1 Understandability

Table 5-1 shows the average understandability of each system as rated by each of the ten listening subjects for the six test sentences. The last row of the table shows the average understandability of each system across all listening subjects.

Table 5-1: Average understandability ratings

Subject	DCS	Ldom	Odom1	Odom 2
Subject 1	1.13	4.67	3.83	3.67
Subject 2	1.00	3.17	3.33	4.17
Subject 3	1.33	4.67	4.33	4.33
Subject 4	1.00	4.33	4.17	4.00
Subject 5	1.17	4.50	3.67	4.00
Subject 6	1.17	4.17	3.50	4.00
Subject 7	1.00	4.67	3.67	3.33
Subject 8	1.17	4.67	3.83	3.67
Subject 9	1.00	4.33	3.67	3.17
Subject 10	1.17	4.67	3.17	3.83
Average	**1.11**	**4.38**	**3.72**	**3.82**

The results shown in Table 1 are plotted in Figure 5.2 to give a graphical representation of differences in understandability between the systems. A rating of three and above

shows the acceptable region of the graph and less than three shows the unacceptable region of the graph.

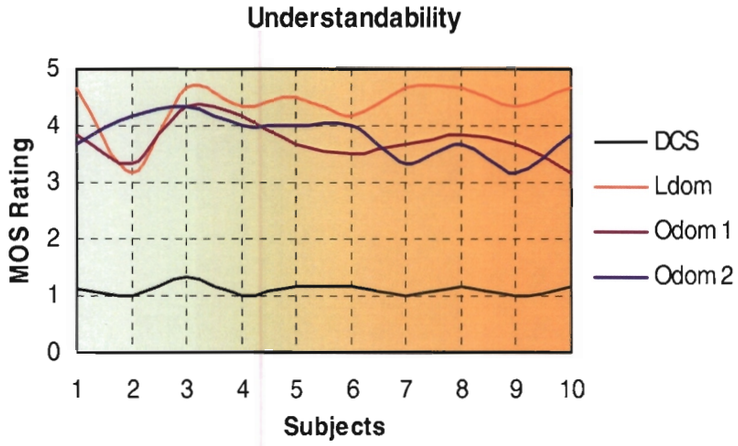


Figure 5.2: Differences in understandability between systems

This is probably the most important requirement of all since it shows how well a synthesized message is understood by the listener after the first time of listening. Table 1 and Figure 5.2 show how well each system measures up to this requirement, which are evaluated as follows:

1. The DCS Afrikaans TTS system has a mean understandability of 1.11. This rating means that the system is not understandable since it requires more than a vast amount of effort to understand the system. Figure 5.2 shows that this system is the worst performer of all systems, and is not in the acceptable region of the graph.
2. The Ldom system has a mean understandability of 4.38. This rating means that the system has little listening effort to no listening effort required to understand the system. Figure 5.2 shows that the system outperforms the other three systems in all cases, except for subject two where it is rated below the two Odom systems.
3. The two Odom systems, Odom1 and Odom2, have a mean understandability of 3.72 and 3.82 respectively. These ratings mean that both systems are in the acceptable region of Figure 5.2, and are close to having little effort required to understand each system. Figure 5.2 also shows that both systems outperform the

DCS system, and that there is not much difference in the understandability of the the two systems.

5.2.2 Naturalness

Table 5-2 shows the average naturalness of each system as rated by each of the ten listening subjects for the six test sentences. The last row of the table shows the average naturalness of each system across all listening subjects.

Table 5-2: Average naturalness ratings

Subjects	DCS	Ldom	Odom1	Odom 2
Subject 1	1.00	4.50	3.67	3.17
Subject 2	1.17	2.67	3.50	3.67
Subject 3	1.50	4.50	3.83	3.83
Subject 4	1.17	5.00	4.83	4.67
Subject 5	1.17	4.00	3.33	3.50
Subject 6	1.17	3.83	3.17	3.17
Subject 7	1.17	4.67	4.00	3.67
Subject 8	1.00	3.67	3.00	3.00
Subject 9	1.00	4.67	3.67	3.17
Subject 10	1.50	4.50	3.17	3.67
Average	**1.18**	**4.20**	**3.62**	**3.55**

The results shown in Table 2 are plotted in Figure 5.3 to give a graphical representation of difference in naturalness between the systems.

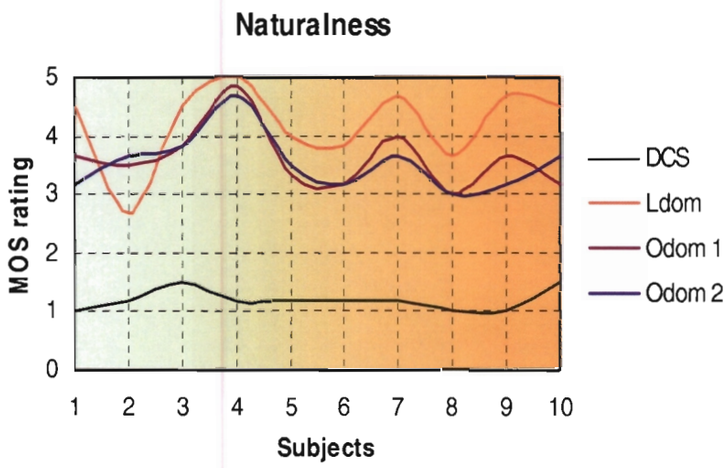


Figure 5.3: Differences in naturalness between systems

This is another very important requirement since it shows well the voice quality of each system compares to that of a human voice. Table 2 and Figure 5.3 show how well each system measures up to this requirement, which are evaluated as follows:

1. The DCS system has a mean naturalness of 1.18. This rating means that the system sounds robotic and rates worst than unnatural. Figure 5.3 also shows that this system performs the worst of all systems, and is in the unacceptable region of the graph.
2. The Ldom system has a mean naturalness of 4.20. This rating means that the system rates between sounding like a human voice and close to sounding like a human voice. Figure 2.3 show that this system outperforms all the other systems except in the case of subject 2.
3. The open domains systems, Odom1 and Odom2, have a mean naturalness of 3.62 and 3.55 respectively. These ratings mean that both systems fall in the acceptable region of the graph and are close to sounding like a human voice. From figure 5.3 it is also easy to see that there is little difference between the two systems, and they both out perform the DCS system again.

5.2.3 Pleasantness

Table 5-3 shows the average pleasantness of each system as rated by each of the ten listening subjects for the six test sentences. The last row of the table shows the average pleasantness of each system across all listening subjects.

Table 5-3: Average pleasantness ratings

Subject	DCS	Ldom	Odom1	Odom 2
Subject 1	1.17	3.67	2.83	3.00
Subject 2	1.83	3.17	3.33	3.50
Subject 3	1.50	4.17	3.67	3.50
Subject 4	1.17	4.00	4.00	3.50
Subject 5	1.17	3.67	3.17	3.17
Subject 6	1.00	3.50	3.33	3.50
Subject 7	1.00	4.00	3.00	3.17
Subject 8	1.00	3.67	3.00	2.83
Subject 9	1.00	3.67	3.00	2.33
Subject 10	1.67	4.00	2.83	3.17
Average	**1.25**	**3.75**	**3.22**	**3.17**

The results shown in Table 3 are plotted in Figure 5.4 to give a graphical representation of differences in pleasantness between the systems.

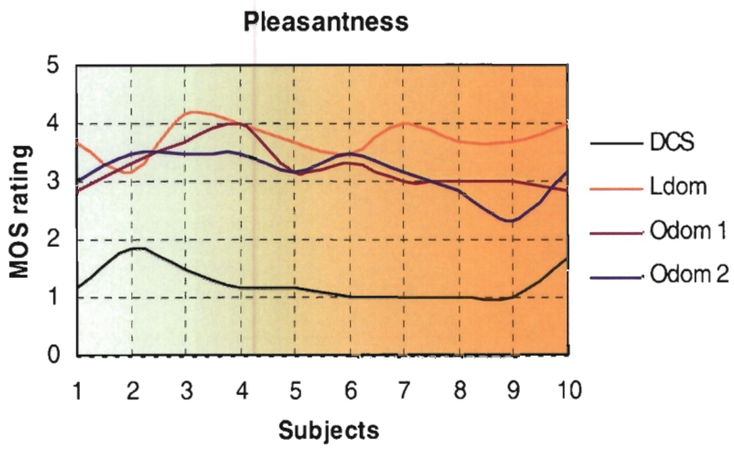


Figure 5.4: Differences in pleasantness between systems

The pleasantness of the systems is a measure of how pleasant the synthetic voice is to listen to, which is an indication on whether or not a listener will be willing to use the system again. Table 3 and Figure 5.4 show how well each system measures up to this requirement, which are evaluated as follows:

1. The DCS system has a mean pleasantness of 1.25. This rating means that the system sounds robotic and unpleasant. Figure 5.4 also shows that this system is the worst performer all the systems, and is in the unacceptable region of the graph.
2. The Ldom system has a mean pleasantness of 3.75. This rating means that the system rates between pleasant and acceptable. Figure 5.4 show that the system outperforms the other systems, except in the case of subject 2 again.
3. The two Odom systems, Odom1 and Odom2, have a mean pleasantness of 3.22 and 3.17 respectively. These ratings mean that the pleasantness of both systems are closer to being acceptable than being pleasant. Figure 5.4 shows that both systems are on the border line of acceptable, especially in the cases of the extreme Afrikaans and English speakers. In two cases both systems rates below acceptable, but on average both systems have acceptable pleasantness.

5.2.4 *Overall impression*

Table 5-1 shows the average overall impression of each system as rated by each of the ten listening subjects for the six test sentences. The last row of the table shows the average overall impression of each system across all listening subjects

Table 5-4: Average overall impression of each system

Subject	DCS	Ldom	Odom1	Odom 2
Subject 1	1.00	3.67	2.83	2.83
Subject 2	1.50	2.83	3.67	3.67
Subject 3	2.00	4.17	3.67	3.67
Subject 4	1.00	3.83	4.00	3.67
Subject 5	1.17	3.50	3.00	3.33
Subject 6	1.17	3.50	3.33	3.00
Subject 7	1.00	4.33	3.33	3.00
Subject 8	1.00	3.67	3.00	3.00
Subject 9	1.00	4.00	3.00	2.33
Subject 10	1.17	4.67	3.33	3.67
Average	**1.20**	**3.82**	**3.32**	**3.22**

The results shown in Table 3 are plotted in Figure 5.5 to give a graphical representation of difference in overall impressions between the systems.

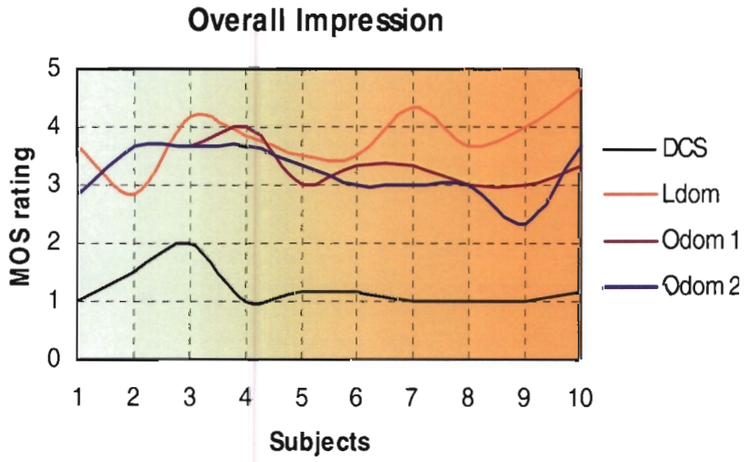


Figure 5.5: Overall impression difference between systems

The overall impression of each system is very important since it indicated whether or not a listener would be prepared to use the system in a real world application. Table 4 and

Figure 5.5 show how well each system rates in terms of the overall impression, which are evaluated as follows:

1. The DCS system has a mean overall impression of 1.20. This rating means that the system rates between being horrible and being poor. Figure 5.5 shows that the system rates below acceptable, and performs of the worst overall impression of all systems.
2. The Ldom system has a mean overall impression of 3.82. This rating means that the system rates closer to being a good system than an acceptable system. On average this systems has the highest overall impression.
3. The two Odom systems, Odom1 and Odom2, have a mean of 3.33 and 3.22 respectively. These ratings mean that both systems again perform above acceptably, and that there is not much difference between the two systems.

5.3 General comments of listening subjects

The final page of the evaluation sheet was used to get the general comments of each system, and to get information on whether or not each listener would be open to the use of the hybrid Ldom/DCS Afrikaans TTS system. This section therefore discusses and evaluates the comments made by the listening subjects. Their views on the hybrid system are also discussed. The full comments can be seen in Appendix C.

The comments on the DCS Afrikaans TTS system were that the system is not understandable, it sounds extremely robotic, it is not pleasant to listen to, it sounds totally unnatural and that the systems is generally a poor system. The comments on the Ldom Afrikaans TTS system were very opposite to that of the DCS system. The comments were that the system is a major step up from the DCS system, it sounds like a human voice, it is very natural, pleasant, had a few audible glitches, but is the preferred system of all systems implemented. The only subject that had a problem with this system was subject 2. This subject said that the system sounds unnatural and that it stutters. From the results discussed earlier, it is clear to see that this subject was more favourable to the two Odom systems. The comments by the other subjects on the two Odom Afrikaans TTS systems were that the systems have very similar voice qualities, meaning that the two

labelling algorithms perform equally well, the voice qualities are not too far off from the Ldom system and that the systems are much better than the DCS system. The audible glitches between words were also noticed and said to decrease the understandability.

When posed with the question of whether or not each subject would be open to the use of the hybrid Ldom/DCS Afrikaans TTS system, eight out of the ten subjects said no. The reasons were that the DCS system is completely unacceptable, and will not complement the Ldom system. Even the two subjects that did said yes were still not in favour of the DCS system, and said that they preferred the Ldom system on its own. Four of the ten subjects said that they would prefer a hybrid Ldom/Odom system instead. Recall from Chapter 3, Section 3.4.2 that an Odom system can be seen as a possible solution to a better back-up system than the DCS system for a hybrid TTS system. These comments prove this idea, and the suggestions made by the listening subjects prompted the design of the system since it will satisfy all the requirements of an advanced TTS system for Afrikaans. The resulting system will have the high understadability, naturalness and pleasantness of the Ldom system integrated with the good voice quality and flexibility of one of the Odom systems. The following section describes the design of this system and what the advantages are over the hybrid Ldom/DCS system.

5.4 The Hybrid Ldom/Odom Afrikaans TTS system

The hybrid Ldom/Odom system for Afrikaans was introduced in Chapter 3 as a possible solution to using a more compatible voice as the back-up voice to the hybrid Ldom/DCS system. The system was not only built as a result of the comments made by the listening subjects, but also to prove that the idea of a hybrid Ldom/Odom system does meet all the requirements of an advanced TTS system. This form of synthesis is however, not available within the framework of the Festival speech synthesis system and therefore the *python*[19] programming language was used to write a new algorithm for achieving this task. The paper entitled, "*An Advanced TTS system for Afrikaans*" [71] was written on the resulting system and can be seen in Appendix E.3. Discussed in the section are the objectives and design of the hybrid Ldom/Odom system for Afrikaans.

[19] Pythom scripting language, http://www.python.org

One downfall of the hybrid Ldom/DCS system as designed by the framework within Festival is that it does not fully meet the objectives of a hybrid system. The system synthesizes the entire utterance with the back-up voice when it is faced with an out of vocabulary word. This is not desired since we want the system to switch between systems to synthesize the out of vocabulary word and then revert back to the Ldom system. The objectives of the new hybrid system are therefore the similar to the objectives shown in Figure 4.4, but difference here is that the system must only switch to the Odom system to synthesize the out of vocabulary words and then return to the Ldom system. This is shown in Figure 5.6, and explained in detail immediately after.

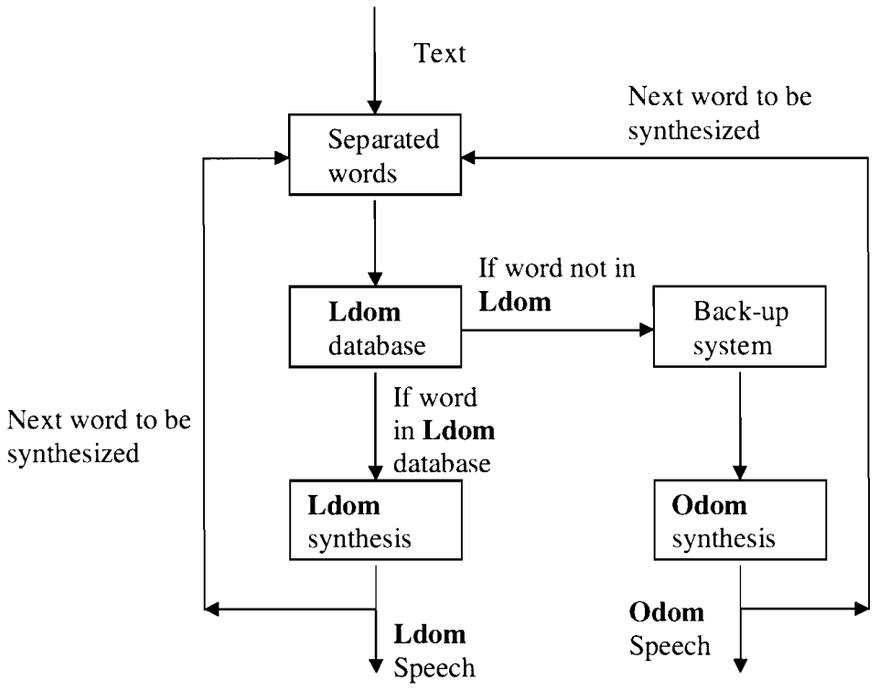


Figure 5.6: Objective of Hybrid Ldom/Odom system

When text is entered into the system each word is send through a decision block to check the occurrence of the word in the Ldom database (dictionary). If the word is in the database then it will be synthesized by the Ldom system. However, if the word is not in the database then it will be synthesized by the Odom system, and after synthesis the system will move on to the next word to be synthesized. The better performing Odom

system, *uct_af_fer_clunits,* was used as the back-up system. The *python* script written to meet the objective described here performs the following steps in order to meets the objectives of Figure 5.6. The script (shown in Appendix *A.5*, co written with the author of [72]) performs the following steps to meet the objective.

- Step 1: Breaks each utterance to be synthesized into words in the order of initial word to final word.
- Step 2: Checks the occurrence of each word in the Ldom vocabulary.
- Step 3: If the word is in the vocabulary then it is stored into a new text file used to store the words that are in the vocabulary before an out of vocabulary word is faced.
- Step 4: When this happens the words that are not in the vocabulary are stored into a text file used to store out of vocabulary words until the next word is faced that is in the vocabulary.
- Step 5: Repeats Step 3 and Step 4 until all words in the utterance to be synthesized is sorted.
- Step 6: Synthesizes the entries of the text files in the order that they were created and concatenates the outputs to form the final output of the synthesized utterance.

An example of this procedure is shown in Figure 5.7 where the system is faced with the out of vocabulary word, *Francois,* in the sentence "*Welkom by Francois se Afrikaans demonstrasie*" (*Welcome to Francois' demonstration of Afrikaans*).

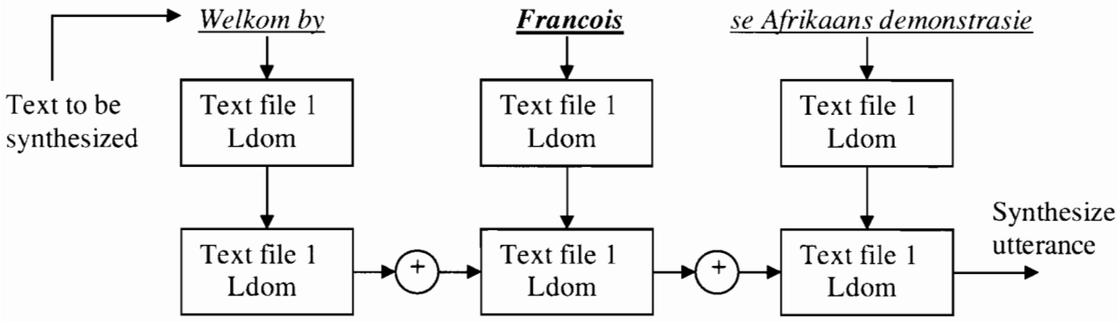


Figure 5.7: New hybrid TTS synthesis procedure [71]

The system was later modified to use a single database for both Ldom and Odom system. The resulting system therefore saves on memory requirements and can meets all the requirements of an advanced TTS system for Afrikaans. The final system has the understandability, naturalness and pleasantness of limited domain synthesis added with the flexibility, acceptable naturalness, pleasantness and understandability of open domain unit selection synthesis. An example of the speech produced by this system is available in Appendix B.5

5.5 Summary

This chapter shows, evaluates and compares the results as per subjective listening test of each of the systems implemented in Chapter 3 in the goal of designing an advanced TTS system for Afrikaans. Results show that the DCS system performs unacceptably, while both the Ldom and Odom perform acceptably in term of the understandability, naturalness and pleasantness required. It was decided to use a hybrid approach to add flexibility to the Ldom by combining it with the high voice quality and flexibility of the first Odom system. The reason for this is that the quality of the Ldom system is ideally what is needed even though the Odom systems perform acceptably. The resulting system meets all the requirements needed to be an advance TTS system and was therefore implemented as the advanced TTS system for Afrikaans.

CHAPTER 6

Conclusions

The goal of this thesis was to design an advanced TTS system for Afrikaans using the Festival speech synthesis system and its techniques of concatenative speech synthesis. The techniques are diphone concatenative synthesis, limited domain and open domain unit selection synthesis. An Afrikaans TTS system was implemented using each of the three techniques, and each system was tested and evaluated using subjective listening tests. This was done to show how well each system measures up to the requirements of an advanced TTS system for Afrikaans. Unfortunately none of the single techniques measured up to all the requirements and therefore, a hybrid approach to TTS synthesis was implemented. This approach adds flexibility to the voice qualities of the Ldom system by using the DCS or one of the Odom systems as a back-up system to synthesize words that are not in the vocabulary of the limited domain system. The hybrid Ldom/Odom Afrikaans TTS system we implemented performed best in terms of meeting all the requirements of an advanced TTS system for Afrikaans. A summary of the work done in achieving this goal is given here and conclusions are made.

6.1 Summary of methodology used to achieve objectives

Advanced TTS systems must have flexibility, understandability, naturalness and pleasantness [5]. Chapter 1 states that the main objective of this study is to design an advanced TTS system for Afrikaans that meets all these requirements. The following is a summary of the methodology used to achieve this objective.

The framework and the concatenative speech synthesis techniques of the Festival speech synthesis system allows for the development of new TTS systems in different languages. For this reason this system was used to achieve the objective at hand. The technique of limited domain unit selection synthesis has the ability produce TTS system with a high degree of understandability, naturalness and pleasantness. The technique can however not provide the flexibility that is required for an advanced TTS system. The first approach was to investigate the technique of diphone concatenative synthesis which has the ability to produce very flexible TTS systems. The DCS Afrikaans TTS system was originally considered to best technique, since it could overcome the problem that the limited domain systems are faced with. The reason for this was that it could produce the flexibility that a limited domain system could not. However, the system had very little understandability, naturalness and pleasantness and was too undesirable to have it implemented as the advanced TTS system for Afrikaans. The poor quality synthesis could be attributed to poor signal analysis. As described in Section 4.1.5, an electroglottograph is required in determining the pitch locations. Since an electroglottograph was not available, an alternative method was to determine the pitch locations manually. This was time consuming as pitchmarks had to be determined for 400 nonsense words in the vocabulary. The pitchmarks from the utterances were then used for LPC analysis, which is required for the re-synthesis of diphones and this contributed to the poor quality of the synthesis obtained for the diphone based system.

We then used the first approach to hybrid TTS synthesis available within the Festival framework. This approach combines the advantages of the limited domain technique with the advantages of the diphone concatenative technique by using the DCS system as a back-up system to synthesize words that are not in the vocabulary of the limited domain system. The hybrid system produced very understandable, natural and pleasant synthesis of the words within the limited domain and showed flexibility by using the DCS system to synthesize out of vocabulary words. The advantage of this system was that any word within the Afrikaans vocabulary could more or less be synthesized. One disadvantage of this system was that the voice quality out of the vocabulary words had a much lower understandability, naturalness and pleasantness than the words within the limited domain.

The reason for this is that quality of the DCS back-up system is too undesirable. Another disadvantage of this system is that it synthesizes the entire utterance with the back-up voice when it is faced with an out of vocabulary word. This was not desired since we wanted the system to switch between systems to synthesize the out of vocabulary word and then revert back to the Ldom system. Because of these two disadvantages this system could not be implemented as the advanced TTS system for Afrikaans. We then investigated the second technique of unit selection synthesis called open domain unit selection synthesis. This technique has the advantage that it can produce TTS systems with a high degree of understandability, naturalness and pleasantness with a similar degree of flexibility as the DCS technique, but has the disadvantage of inconsistency. The results and evaluations of the subjective listening tests show that both the Odom Afrikaans TTS systems implemented performs acceptably in terms of the requirements. The general comments made by the listening subjects however states that the audible glitches experienced between words in cases, made the systems inconsistent. For this reason neither of the two Odom systems for Afrikaans was implemented as the advanced TTS system for Afrikaans.

The final approach was a new hybrid system that adds flexibility to the Ldom system by using one of the Odom systems as a back-up system to synthesize out of vocabulary words. This form of synthesis, shown in Figure 5.6, is not available within the Festival framework and was developed using the *python* programming language. The algorithm ensures that this system switches to the back-up system only to synthesize the out of vocabulary word and then reverts back to the Ldom system. The final system has the understandability, naturalness and pleasantness of the Ldom system backed up with the acceptable understandability, naturalness and pleasantness of the best performing Odom system. The hybrid Ldom/Odom Afrikaans TTS system was then implemented as the advanced TTS system for Afrikaans.

6.2 Conclusions

Based on the results and the evaluations of the results, the following conclusions were drawn.

1. The diphone concatenative Afrikaans TTS system can not be implemented as the advanced TTS system for Afrikaans since the system has an unacceptable degree of understandability, naturalness and pleasantness as shown in Figure 6.1.

2. The limited domain unit selection Afrikaans TTS system produces the high degree of understandability, naturalness and pleasantness that is required. See Figure 6.1. The system however, has a limited flexibility and therefore can not be implemented as the advanced TTS system for Afrikaans.

3. The hybrid Ldom/DCS Afrikaans TTS system meets all the requirements of an advanced TTS system but can not be implemented because of the unacceptable voice quality of the DCS system.

4. The two open domain unit selection Afrikaans TTS systems produced similar results and both performed acceptable in terms of the requirements needed. See Figure 6.1. However, neither of the two can implemented as the advanced TTS system for Afrikaans for the reason that the audible glitches experienced between words makes the systems inconsistent and decreases understandability in cases.

5. The second hybrid approach to TTS synthesis adds flexibility to the Ldom system by using the DTW labeled Odom system to synthesize out of vocabulary words. This system meets all the requirements of an advanced TTS system and is therefore implemented as the advanced TTS system for Afrikaans. This new hybrid technique, not available within the framework

of Festival, can also be used for the design of advanced TTS systems for languages.

The techniques and TTS systems constructed in this thesis were built in test environments using a non-professional speaker. Results can be further improved with use of profession recording equipment, profession recording studios and a professional speaker.

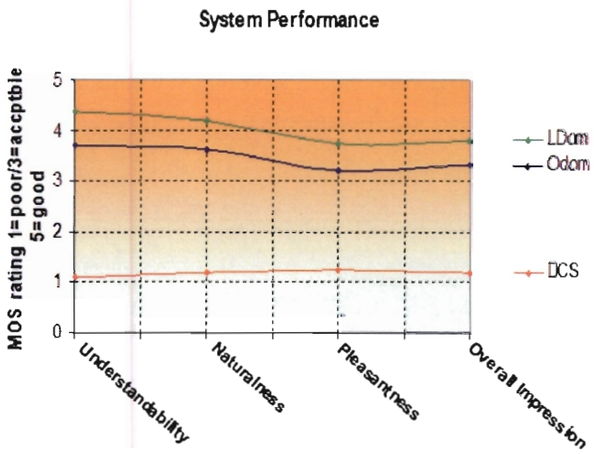


Figure 6.1: System Performances

6.3 Recommendations

For the design of advanced TTS systems in different languages we recommend the use of the hybrid Ldom/Odom TTS technique. This technique combines the advantages of both techniques into one system that meets all the requirements of an advanced TTS system. We recommend the use of a single database for both systems, recorded in a professional environment by a professional speaker. For most effective results this database should be a list of the most commonly used words within a language. This will ensure that the system rarely faces out of vocabulary words, and when the system is faced with an out of vocabulary word, the above acceptable speech quality of the Odom synthesis (See Figure 6.1) will be used to synthesize the word. Future work on these systems can be done on intonational and prosody modeling to add more naturalness, and flow to the synthetic speech produced by the system.

Bibliography

[1] L. Lerato, "*Hierarchical Methods for Large Population Speaker Identification using Telephone Speech*", Master's Thesis: University of Cape Town, South Africa, 2003

[2] M. Huckvale, "*Speech synthesis, speech simulation and speech science*", in Proceedings of ICSLP 2002, pp 1261 – 1264, 2002

[3] S. Lemmetty, "*Review of Speech Synthesis Technology*", Master's Thesis: Helsinki University of Technology, Turkey, 1999

[4] J. L. Flanagan, "*Voices of men and machines*", Journal of Acoustical Society of America, Volume 51, pp 1375 – 1387, 1972

[5] F. Rousseau, D. J. Mashao, "*A Hybrid Text-To-Speech System for Afrikaans*", in Proceedings of SATNAC 2005, 2005

[6] F. Rousseau, D. J. Mashao, "*Increased Diphone Recognition for an Afrikaans TTS system*", in Proceeding of PRASA 2004, pp 113 – 117, 2004

[7] A. W. Black, K. Lenzo "*Building Synthetic Voices*", unpublished document, Carnegie Mellon Universtiy, Available at http://festovx.org.bsv

[8] N. Rochford, "*Developing a new voice for Hiberno-English in the Festival Speech Synthesis System*", Undergraduate Thesis Project: Trinity College, Dublin, available at http://www.cs.tcd.ie/courses/csll/projects4.html, 2003

[9] A. W. Black, R. Clark, K Richmond, S King, "*The Festival Speech Synthesis System*" University of Edinburgh, Scotland, available at: www.csrt.ed.ac.uk/projects/festival

[10] A. Conkie, "*Robust unit selection system for speech synthesis"*, in Proceedings of EUROSPEECH 1999, 1999

[11] R. Clark, K. Richmond, S. King, "*Festival 2 - Build your own general purpose unit selection speech synthesizer*", in Proceedings of 5th ISCA workshop on speech synthesis 2004, pp 173 – 178, 2004

[12] A. Hunt, A. W. Black, "*Unit selection in a concatenative speech synthesis system using a large speech database*", in Proceedings of ICASSP 1996, Volume 1, pp 373 – 376, 1996

[13] A. W. Black, K. A. Lenzo, "*Limited domain synthesis*", in Proceedings of ICSLP 2000, Volume 2, pp 411 – 414, 2000

[14] A. W. Black, K. Lenzo, "*Optimal Data Selection for Unit Selection Synthesis*", in Proceeding of the ISCA 4th Speech Synthesis Workshop 2001, pp 63 – 67, 2001

[15] CENSUS 2001, "*Statistics South Africa*", online resource, available at: http://www.statssa.gov.za/census01/html/default.asp, last accessed April 2005

[16] J. Oliver "*Afrikaans*", online resource, available at: http://www.cyberserv.co.za/users/ /~jako/lang/afr.htm, last accessed 4 October 2005

[17] J. C. Wells, "*SAMPA computer readable phonetic alphabet*", online resource, available at: www.phon.ucl.ac.uk/home/sampa/home.htm, last accessed 04 October 2005

[18] D. Wissing, J. P. Martens, U. Janke, W Goedertier, "*A Spoken Afrikaans Language Resource Designed for Research on Pronunciation Variations*", in Proceedings of LREC 2004, pp 669-672, 2004

[19] D. Wissing, "*Protokol vir Bree Fonetiese Transkripsie vir Afrikaans*", online resource, available at: http://www.puk.ac.za/HLT_Resources/Corpora, last accessed 7 August 2005

[20] D. Wissing, "*Speech Assessment Methods Phonetic Alphabet – Afrikaans Version*", online recourse, available at: www.phon.ucl.ac.uk/home/sampa/afrikaans-draft.htm, last accessed 5 October 2005

[21] M. J. Wagner, "*Synthesizing intelligible speech from Afrikaans*", Doctoral Thesis: University of Port Elizabeth, South Africa, 1995

[22] J. Roux, L. Botha, J du Preez, "*African Speech Technology*", online resource, available at: www.ast.sun.ac.za, last accessed 5 October 2005

[23] J. Kominek, C. Bennett, A. W. Black, "*Evaluating and Correcting Phoneme Segmentation for Unit Selection Synthesis*", in Proceedings of EUROSPEECH 2003, pp. 313–316, 2003

[24] D. Klatt, "*Review of Text-To-Speech Conversation for English*", Journal of the Acoustic Society of America, Volume 82, pp 727 – 793, 1987

[25] "*UK Telephone History*", online resource, available at: http://web.ukonline.co.uk/freshwater/histuk.htm, last accessed 20 September 2005

[26] J. Kivimäki, "*Very low bit rate speech coding using speech recognition, analysis and synthesis*", Master's Thesis: Tampere University of Technology, Finland, 2000

[27] A. Pasanen, "*Speech Synthesis*", unpublished document, available at: http;//www.cs.tut.fi/sgn/arg/synteesi/pasanen.pdf, 2001

[28] C. H. Shadle, R. I. Damper, "*Prospects for articulatory synthesis: A position paper*", in Proceedings of the 4th ITRW on Speech Synthesis 2001, pp 121 – 126, 2001

[29] G. Rosen, "*A Dynamic Analog Speech Synthesizer*", Journal of the Acoustical Society of America, Volume 30, pp 201 – 209, 1958

[30] A. Iida, "*A Study on Corpus-based Speech Synthesis with Emotion*", Master's Thesis: Graduate School of Media and Governance, Keio University, Japan, 2002

[31] I. Mattingly, "*Speech Synthesis for Phonetic and Phonological Models*", in *Current Trends in Linguistics*, Editor: Thomas A. Sebeok, Volume 12, Mouton, The Hague, pp. 2451-2487, 1974.

[32] Y. Sagisaka, N Kaiki, N. Iwahashi, K. Mimura, "*ATR ν-Talk speech synthesis system*", in Proceedings of ICSLP 1992, pp 483 – 486, 1992

[33] K. Wooil, "*Speech Production*", online resource, Available at: http://ispl.korea.ac.kr/~wikim/research/speech.html, Last accessed 6 October 2005, Korea University, Korea

[34] G. Fant, "*Acoustic theory of speech production*", Mouton, The Hague, 1960

[35] J. P. Hornak, "*The Basics of MRI*", online book, available at: http://www.cis.rit.edu/htbooks/mri

[36] D. Klatt, "*Software for a Cascade/Parrallel Formant Synthesizser*", Journal of the Acoustic Society of America, Volume 67, pp 971 – 995, 1980

[37] J. Adell, A. Bonafonte, "*Towards phone segmentation for concatenative speech synthesis*", in Proceeding of the 5th ISCA Speech Synthesis Workshop 2004, pp 139 – 144, 2004

[38] D. Chappell, J. H. L. Hansen, "*Spectral Smoothing for Concatenative Speech Synthesis*", in Proceedings of ICSLP 1998, Volume 5, pp 1935 – 1938, 1998

[39] Oxford University Press, "*The Concise Oxford Dictionary*", Ninth edition, United States, Oxford University Press, New York

[40] J. Olive, J. van Santen, B. Moebius, C. Shih, "*Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*", Editor: Richard Sproat, Kluwer Academic Publishers, Norwell, pp 191 – 228, 1998

[41] Y. Saikachi, "*Building a Unit Selection Voice for Festival*", Master's thesis: University of Edinburgh, Scotland, 2003

[42] S. Kishore, A. W. Black, "*Unit Size in Unit Selection Speech Synthesis*", in Proceeding of EUROSPEECH 2003, pp 1317 – 1320, 2003

[43] M. Beutnagel, M. Mohri, M. Riley, "*Rapid unit selection from a large speech corpus for concatenative speech synthesis*", in Proceedings of EUROSPEECH 1999, pp 607 – 610, 1999

[44] M. Beutnagel, A. Conkie, A .K. Syrdal, "*Diphone Synthesis using Unit Selection*", in Proceedings of the 3rd ESCA/COCOSDA International Workshop on Speech Synthesis 1998, pp 185 – 190, 1998.

[45] T. Dutoit, "*Introduction to Speech Synthesis Systems*", Kluwer Academic Publishes, Dordrecht, 1997

[46] A. W. Black, R. Clark, K Richmond, S King, "*The Festival Speech Synthesis System: System Documentation*", Edition 1.4, Released June 1999, available at: www.csrt.ed.ac.uk/projects/festival/manual

[47] P. Taylor, A. W. Black, R. Caley, "*The architecture of the Festival speech synthesis system*", in Proceedings of the third ESCA Workshop in Speech Synthesis, pp. 147 – 151, 1998

[48] P. Taylor, R. Caley, A. W. Black, S. King, "*Edinburgh Speech Tools library: System Documentation*", Edition 1.2.0, Released June 1999, available at: http://www.cstr.ed.ac.uk/projects/speechtools/manual-1.2.0

[49] J. Allen, S. Hunnicut, D. Klatt. "*From Text to Speech: the MITalk System*", Cambridge University Press, 1987.

[50] W. N. Campbell, S. D. Isard, A. I. C. Monaghan, J. Verhoven, "*Duration, pitch and diphones in the CSTR TTS system*", in Proceedings of ICSL 1990, pp 825-828, 1990

[51] M. Vegnaduzzo, "*Modeling Intonation for the Italian Festival TTS using Linear Regression*", Master's Thesis: University of Edinburgh, Scotland, 2003

[52] CMU Speech Software, "*Festvox*", available at: http://festvox.org/, Last accessed 3 October 2005

[53] O. Salor, B. Pellom, M. Demirekler, "*Implementation and Evaluation of a Text-to-Speech Synthesis System for Turkish*", in Proceedings of INTERSPEECH-2003/Eurospeech-2003, pp. 1573 - 1576, 2003

[54] K. Sjolander, J Beskow, "*Wavesurfer*", Open Source sound visualization and manipulation tool, available at: www.speech.kth.se/wavesurfer, last accessed 25 September 2004

[55] D. T. Chappell, J. H. L. Hansen, "*A comparison of spectral smoothing methods for segment concatenation based speech synthesis*", Speech Communication, Volume 36, pp 343 – 374, 2002

[56] B. Langer, A. W. Black, "*Creating a Database of Speech in Noise for Unit Selection Synthesis*", in Proceedings of the fifth ISCA Speech Synthesis Workshop, pp 229 – 230, 2004

[57] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Möbius, B. Säuberlich, "*Restricted Unlimited Domain Synthesis*", in Proceedings of EUROSPEECH 2003, pp 1321 – 1324, 2003

[58] M. S. B Kritzenbeurg, "*Groot Woordeboek*", Pretoria, Vanschaik 1972

[59] M. J. Makashay, C. W. Wightman, A. K. Syrdal, and A. Conkie, "*Perceptual evaluation of automatic segmentation in Text-to-Speech synthesis*", in Proceedings of ICSLP 2000, pp 431 – 434, 2000

[60] F. Malfrere, O. Deroo, T. Dutoit, C Ris, "*Phonetic alignment: speech synthesis-based vs. viterbi-based Source*", Speech Communication, Volume 40, Issue 4, pp 503 – 515, 2003

[61] S. G. Paulo, L. C. Oliveira, "*DTW-based Phonetic Alignment Using Multiple Acoustic Features*", in Proceedings of EUROSPEECH 2003, pp 309 – 312, 2003

[62] Carnegie Mellon University, "*The CMU Sphinx project page*", online resource, available at: http://cmusphinx.sourceforge.net/webpage/html/download.php#SphinxTrain, last accessed 22 September 2005

[63] Carnegie Mellon University, "*Sphinx-II User Guide*", online resource, available at: http://cmusphinx.sourceforge.net/sphinx2/, last accessed 22 September 2005

[64] J. Kominek, C. Bennett, A. W. Black, "*Evaluating and Correcting Phoneme Segmentation for Unit Selection Synthesis*", in Proceedings of EUROSPEECH 2003, pp 313 – 316, 2003

[65] H. Escalona, O. Villagomez, I. Kirschning, "*Estimation of Duration Models for Phonemes in Mexican Speech Synthesis*", ICSLP 2000, pp 685 - 688, 2000

[66] Media 24 Digital, "*Rapport, Afrikaans online newspaper*", online resource, available at: http://www.news24.com/Rapport/Home/0,,,00.html, updated daily, last accessed 4 October 2005

[67] Wikipedia, "*The free encyclopedia - Afrikaans*", online resource, available at: http://af.wikipedia.org/wiki/Suid-Afrika, last accessed 17 September 2005

[68] B. L. Appanna, M. Skosan, D. J. Mashao, "*Using high-level and low-level feature concatenation for speaker identification*", in Proceeding of PRASA 2004, pp 103 - 106, 2004

[69] A. Raux, A. W. Black, "*A Unit Selection Approach to F0 Modeling and ItsApplication to Emphasis"*, in Proceedings of IEEE ASRU 2003, 2003

[70] M. Viswanathan, M. Viswanathan, "*Measuring speech quality for text-to-speech systems:development and assessment of a modified mean opinion score MOS) scale*", Computer Speech & Language, Volume 19, pp 55-83, 2005

[71] F. Rousseau, D. J. Mashao, "*An advanced TTS System for Afrikaans*", Submitted to PRASA 2005

[72] M. Skosan, "*Histogram Equalization for Robust Text-Independent Speaker Verification in Telephone Environments*", Master's Thesis: University of Cape Town, South Africa, 2005