



University of Fort Hare
Together in Excellence

Development of isiXhosa Text-To-Speech Modules to Support e-Services in Marginalized Rural Areas

by

Siphe Mhlana

A thesis submitted in fulfillment of the requirements for the degree

Master of Science

In the

Department of Computer Science

University of Fort Hare

Supervisor: Dr. Okuthe P. Kogeda

Co-Supervisor: Dr. M. Thinyane

Submitted: December 2011

Declaration

I, Sophe Mhlana (Student Number: 200600591), the undersigned, hereby declare that the work contained in this dissertation is my own original work and has not previously been submitted at any educational institution for a similar or any other degree. Information extracted from other sources is acknowledged accordingly.

Signature.....

Date.....

Acknowledgements

I would like to thank God who has made it possible for me to undertake and complete this study; I can do all things through Christ who strengthens me.

I would like to thank Dr. M. Thinyane, Dr. O.P. Kogeda, Mr. S. Ngwenya and Mr. M.S. Scott for the support they gave me during the development of the application and while writing the dissertation. Without you next to me it would not be possible to complete these two years of Masters.

I would also like to thank *Telkom SA* for financial sponsorship throughout my education at Fort Hare University.

To my family: Mr. R.S Mhlana, Ms. S.S Mhlana, Ms. O Mhlana, Mr. M. S Mhlana and Mrs. N Soldati, thank you very much for your understanding and support throughout my education. I know that it was not easy, especially since I was not able to spend my university holidays with you, due to my commitment to this project, but you understood and supported me. Thank you a million times.

I would also like to thank Mr. Saul from African languages for assisting me with some of the isiXhosa words and Ms. L Mohasi from the Stellenbosch Department of Engineering for assisting me in the development of the application.

I would like to thank Marclie Davel and Aby Louw from CSIR for assisting me during the development of the application.

Not forgetting my girlfriend, Ms Asisipho Dyani, for always encouraging me to do my best throughout the duration of this difficult research period.

Let me thank my friends, Mr. P Giwu, Mr. A. Mditshwa, Miss S. Mali and Mr. X. Jevu for all the help and advice that you gave me throughout my research.

Not forgetting my church mates: Mr. M. Nako, Mr. T. Dube, Mr. Jayiya, Mr. L. Moyo, Mr. R. Macheke and Mr. C. Bell, thank you for your support during difficulties, may the good Lord richly bless you and grant you all your wishes in life.

I would also like to thank my classmates, Wandile, Duduzile, Makaziwe, Nombulelo, and everyone else who assisted in the completion of my research.

I would also like to thank Mr. N. Jere “I am *fine too...andifuni kubhala mna*” for assisting me during the writing of my dissertation.

Finally, I would like to thank all those who supported me up to this stage especially in the successful completion of this research. Thank you all.

Dedication

This work is dedicated to my late mother, Eunice, and my late father, Tickens Mhlana. They have been a source of inspiration from my childhood and have encouraged me, throughout my life, to reach greater heights. Thank you, I still love you. This work also dedicated to my secondary school teacher, Mrs. G. Mbedu, who has shown me the direction to success; without you telling me that I can do better in my studies if only I focus, it would have been impossible for me to reach this point in life.

Publications

Aspect of this research have been published in the following conference papers:

Mhlana, S; Thinyane, M. & Kogeda, P, Okuthe. (2010). *Developing Xhosa-Audio Interface for e-Service in Marginalized Rural Areas: A case study*. SATNAC Conference 2010, 05 September - 08 September 2010, Spier Wine Estate, Stellenbosch, Western Cape.

Mhlana, S; Dhir, A; Kaur, P; Ylä-Jääski, A; Kujala, S; Jere, N; Ngwenya, S. (2012). *Exploring Design Ideas for Rural Users: A Cross-Cultural Approach focusing on Communities*. The 5th International Conference on Advances in Computer-Human Interactions: ACHI 2012, January 30 – February 04, 2012, Valencia, Spain.

Acronyms and Abbreviations

ANSI	American National Standard Institute
ASCII	American Standard Code for Information Interchange
COFISA	Cooperating Framework on Innovation System between Finland and South Africa
CSTR	Center for Speech Technology Research
DECTalk	Digital Equipment Corporation
DTMF	Dual Tone Multi Frequencies
GUI	Graphical User Interface
HMM	Hidden Markov Models
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
ICT4D	Information and Communication Technology for Development
IT	Information Technology
LPC	Liner Predictive Coding
LTS	Letter-To-Sound
OECD	Organization for Economic Co-operation and Development
PSOLA	Pitch-Synchronous Overlap and Add
SLL	Siyakhula Living Lab
THRIP	Technology and Human Resources for Industry Programme
TTS	Text-To-Speech
URL	Uniform Resource Locator
VODER	Voice Operating Demonstrator

Abstract

Information and Communication Technology (ICT) projects are being initiated and deployed in marginalized areas to help improve the standard of living for community members. This has led to a new field, which is responsible for information processing and knowledge development in rural areas, called Information and Communication Technology for Development (ICT4D). An ICT4D project has been implemented in a marginalized area called Dwesa; this is a rural area situated in the wild coast of the former homeland of Transkei, in the Eastern Cape Province of South Africa. In this rural community there are e-Service projects which have been developed and deployed to support the already existent ICT infrastructure. Some of these projects include the e-Commerce platform, e-Judiciary service, e-Health and e-Government portal. Although these projects are deployed in this area, community members face a language and literacy barrier because these services are typically accessed through English textual interfaces. This becomes a challenge because their language of communication is isiXhosa and some of the community members are illiterate. Most of the rural areas consist of illiterate people who cannot read and write isiXhosa but can only speak the language. This problem of illiteracy in rural areas affects both the youth and the elderly.

This research seeks to design, develop and implement software modules that can be used to convert isiXhosa text into natural sounding isiXhosa speech. Such an application is called a Text-to-Speech (TTS) system. The main objective of this research is to improve ICT4D eServices' usability through the development of an isiXhosa Text-to-Speech system. This research is undertaken within the context of Siyakhula Living Lab (SLL), an ICT4D intervention towards improving the lives of rural communities of South Africa in an attempt to bridge the digital divide.

The developed TTS modules were subsequently tested to determine their applicability to improve eServices usability. The results show acceptable levels of usability as having produced audio utterances for the isiXhosa Text-To-Speech system for marginalized areas.

Keywords: ICT, ICT4D, Text-to-Speech System, isiXhosa, illiterate, SLL, Marginalized Rural Area, Usability

Table of Contents

1	CHAPTER 1: INTRODUCTION	14
1.1	Introduction	14
1.2	Text-To-Speech Technology	14
1.3	Research Context and Background	15
1.4	Research Problem	16
1.5	Research Question	17
1.6	Objectives of the Research	19
1.7	Research Methodology	22
1.8	Motivation of Research	23
1.9	Contribution of the Thesis	23
1.10	Structure of the thesis	24
1.11	Conclusion	24
2	CHAPTER 2: LITERATURE REVIEW	25
2.1	Introduction	25
2.2	Overview of Text-to-Speech.....	25
2.3	Text To Speech Synthesis	29
2.4	Text-to-Speech Architecture	30
2.4.1	Text	30
2.4.2	Text Analysis	30
2.4.3	Phonetic Analysis	31
2.4.4	Prosodic Analysis	33
2.4.5	Speech Synthesis	33
2.5	Text-to-Speech Engines	33
2.5.1	Festival Speech Engine	33
2.5.2	Flite Speech Engine	34
2.5.3	eSpeak Speech Engine	34
2.6	Applications of Speech Synthesis	35
2.7	ICT in Marginalized Rural Areas	36
2.8	ICT for Development (ICT4D) in Developing Countries	39
2.9	ICT Usage in Marginalized Rural Areas	42
2.10	Study Area: Dwesa.....	43

2.11	Overview of IsiXhosa	46
2.12	Developing the IsiXhosa Text-to-Speech System	48
2.13	Related Work	49
2.13.1	Commercial.....	49
2.13.2	Academic	52
2.14	Conclusion	54
3	CHAPTER 3: SYSTEM DESIGN AND ARCHITECTURE.....	55
3.1	Introduction	55
3.2	System Back-end Design.....	55
3.3	Database Design on Festival	56
3.3.1	User Roles	58
3.3.2	Administrator.....	58
3.3.3	Community Members.....	59
3.3.4	Students.....	60
3.4	System Architecture	61
3.4.1	Filter Module	62
3.4.2	IsiXhosa Vocabulary.....	63
3.4.3	Word Segmentation Module.....	63
3.4.4	Phrasing Module.....	63
3.4.5	Generation Module	63
3.5	System Front-end Design	64
3.6	Front-end System Architecture.....	64
3.7	Design Considerations of the System.....	65
3.8	Conclusion.....	65
4	CHAPTER 4: IMPLEMENTATION	66
4.1	Introduction	66
4.2	System Implementation	66
4.3	Text-To-Speech System Stages	66
4.4	Recording of a Voice	66
4.4.1	Speaker	67
4.4.2	Recording Environment.....	67
4.5	Database	67
4.6	Database Table.....	69
4.6.1	Words Table.....	69

4.6.2	Sentence Table	70
4.6.3	Month Table	71
4.6.4	Days of the Week Table.....	71
4.6.5	Time Table	72
4.7	System Administrator	73
4.8	Festvox	74
4.8.1	Phonset Module	75
4.8.2	Letter-To-Sound Module	77
4.8.3	Phrasing Module.....	78
4.8.4	Intonation Analysis	80
4.8.5	Duration Parameter Module	81
4.8.6	Fundamental Frequency (F0) Generation	82
4.8.7	Waveform Generation Module	83
4.9	Festival Application.....	84
4.10	Community Members using the system	85
4.11	Conclusion	85
5	CHAPTER 5: SYSTEM TESTING AND RESULTS	86
5.1	Introduction	86
5.2	Testing Apparatus	86
5.3	Testing Environment	86
5.4	Functionality Testing	86
5.4.1	Components testing	87
5.4.2	Usability Testing	89
5.4.2.1	System users	89
5.4.2.2	Training needs	90
5.4.2.3	Text-To-Speech Conversion.....	91
5.4.2.4	Results of Text-To-Speech conversion exercise	93
5.5	Training of the System	94
5.6	Conclusion.....	99
6	CHAPTER 6: SUMMARY, DISCUSSION AND CONCLUSION	100
6.1	Introduction	100
6.2	Summary of the Research.....	100
6.3	Achievements of the objectives.....	101
6.4	Problems Encountered.....	104

6.5	Future Work.....	105
6.6	Discussion of Results	106
6.7	Conclusion.....	107
7	References.....	108
8	Appendix A – Technologies Required	116
8.1	Operating System	116
8.2	Gcc.....	117
9	Appendix B – System Implementation.....	118
9.1	Phonset Module	118
9.2	Letter-to-Sound Rule	122
9.3	Phrasing Module.....	125
10	Appendix C – Database.....	127
10.1	Words Table	127
10.2	Sentence Table	127
10.3	Voice database.....	128
11	Appendix D – Usability and Functionality Testing.....	130
11.1	Usability testing – Questionnaires Part 1.....	130
12	Usability testing – Questionnaires Part 2.....	134
12.1	System usability testing	134

List of Figures

Figure 2-1: Text-To-Speech Architecture.....	32
Figure 2-2: South Africa Provinces (Statistic South Africa, 2006).....	47
Figure 3-1: Front-end and back-end Architecture.....	56
Figure 3-2: Database Model.....	57
Figure 3-3: Administrator Use Case Diagram.....	59
Figure 3-4: Community Use Case Diagram.....	60
Figure 3-5: Students Use Case Diagram.....	61
Figure 3-6: System Architecture.....	61
Figure 3-7: General outline of TTS System.....	64
Figure 4-1: Voice path.....	84
Figure 4-2: The error when you are playing music on the background.....	84
Figure 4-3: Festival Commands.....	85
Figure 5-1: Path where festival voices are stored.....	87
Figure 5-2: Location of all festival voices.....	87
Figure 5-3: Festival testing.....	88
Figure 5-4: Festival output.....	88
Figure 5-5: Displaying voice options.....	88
Figure 5-6: Festival text conversion.....	88
Figure 5-7: Festival changing of voice speed.....	89
Figure 5-8: Training needs assessment chart.....	91
Figure 5-9: System Rating Assessment Chart.....	94
Figure 5-10: System Training Results for Words.....	96
Figure 5-11: System Training Results for Sentences.....	98
Figure 5-12: Combined Results of the System Training.....	99

List of Tables

Table 1-1: Summary of Questions, Objectives and Methodology	21
Table 2-1: South African Departments in ICT.....	41
Table 2-2: IsiXhosa pronunciation.....	46
Table 2-3: IsiXhosa clicks consonant.....	46
Table 4-1: Wards table.....	70
Table 4-2: Sentence table.....	71
Table 4-3: Month table.....	71
Table 4-4: Days of the week.....	72
Table 4-5: Time table.....	73
Table 5-1: Sample users	89
Table 5-2: Sample isiXhosa words used by users	91
Table 5-3: System rating results.....	93

Table of Listings

Listing 4-1: Database connection module.....	69
Listing 4-2: System Administrator.....	73
Listing 4-3: Listing that links other modules.....	74
Listing 4-4: Phonetset listing for isiXhosa language.....	76
Listing 4-5: Letter-To-Sound mapping rules.....	78
Listing 4-6: CART Tree Module.....	79
Listing 4-7: Phrasing Module with types of Breaks.....	79
Listing 4-8: Determination of the tone from vowels and consonants.....	80
Listing 4-9: Zscores tree prediction of duration.....	81
Listing 4-10: Duration module that links to the zscores tree.....	82
Listing 4-11: Fundamental Frequency Module code.....	83
Listing 5-1: Sample isiXhosa sentences.....	992

1 CHAPTER 1: INTRODUCTION

1.1 Introduction

In this Chapter, the Text-To-Speech technology is reviewed in Section 1.2. In Section 1.3, the research context and background are presented. In Section 1.4, the research problem is outlined. In Section 1.5, the research questions are explored in details. In Section 1.6, the objectives of the research are discussed in detailed. In Section 1.7, the methodology followed in this research will be outlined. In Section 1.8, the research motivation will be presented. In Section 1.9, the contribution of the thesis will be presented. In Section 1.10, thesis organization is presented and the chapter is concluded in Section 1.11.

1.2 Text-To-Speech Technology

Text-To-Speech (TTS) systems are widely used to generate spoken utterances from text (Bickley et al., 1998). This application can be used to render text through digital audio. Most speech modules can be categorized by the method they use to translate phonemes into audible sound (Parssinen, 2007).

The speech modules used are commercially available in English (e.g. British, United States), some Indian languages (e.g. Hindi, Tamil and Urdu), French, Spanish, and Swahili. However, there are numerous speech module synthesizers used all over the world but, as far as we are aware, none exists in isiXhosa (Black et al., 1998). This suggests that there is a need to design, develop and implement a Text-To-Speech system for people living in marginalized areas of the Eastern Cape where most isiXhosa speakers live.

There are similar projects which are already deployed in marginalized areas in order to improve the standard of living of the inhabitants. These projects are part of the Information and Communication Technology for Development (ICT4D) effort, which is used as a catalyst for the development of marginalized areas. The problem in these projects is that they are all written in English so it is difficult for illiterate people to use them because, in rural areas, the majority of people from the community cannot even read and write in their mother tongue, which is

isiXhosa. It is difficult for isiXhosa speakers living in rural areas to access information from the internet because of the language barrier. Even on the internet there is not much information that is written in isiXhosa to accommodate such people. There is a need to develop an application, for the isiXhosa language, which can improve the use of computers in rural areas, particularly in the Eastern Cape Province. The application that will help illiterate people in rural areas is called a Text-To-Speech system; this system can be used to convert isiXhosa text into isiXhosa speech.

In order to implement and design the appropriate Text-To-Speech system, there are different types of speech engines that can be used in its development. In this study, the researcher proposed the use of Festival free open-source software to implement the system because it allows every language to be integrated into it. Festival open-source software is written in C++, which is a stable and portable multilingual speech synthesis framework. Festival offers a general framework for building speech synthesis systems and including examples of various modules. It was developed by Alan Black at the Center for Speech Technology Research (CSTR) of the University of Edinburgh (Black et al., 1998). The reason why Festival open-source was chosen is that the code can be modified and is affordable for rural areas.

There are modules which are required to be included when the development of the Text-To-Speech system is being implemented under Festival synthesis for a new language, such as: phoneset, phrasing, intonation, duration, and waveforms synthesis. The isiXhosa language also involves vowels and consonants that need to be considered during the implementation of the system.

1.3 Research Context and Background

This research is undertaken in the context of the deployment of ICT4D projects in marginalized communities. It is part of a bigger project called Siyakhula Living Lab. Siyakhula Living lab is an ICT4D initiative which aims to provide rural communities with an ICT platform to enhance communication, thereby opening up for other developmental initiatives. The project is situated in Dwesa, a rural marginalized community in the Eastern Cape region within the Mbashe municipality. Siyakhula Living Lab covers sixteen schools as access centers within a radius of 40 km. Several applications have been developed and deployed on the network to foster

communication amongst the poor disadvantaged community members. Some of the projects which have been developed, or are being developed, include:

- ❖ An e-Government platform which aims to provide a link between the government and its citizens through ICT platforms (Jakachira et al., 2009);
- ❖ An e-Commerce platform with the aim of providing rural communities with a means of marketing and selling their arts and craft products on the world market (Dyakalashe et al., 2009);
- ❖ An e-Judiciary service which delivers online Judiciary services to the Dwesa community (Scott et al., 2010);
- ❖ A helpdesk system for the collaborative sharing of knowledge for the continuous assessment of project activities in Dwesa (Makombe et al., 2011);
- ❖ An e-Health portal for the Dwesa community to access health information and facilities online (Hlungulu et al., 2010);
- ❖ A User Driven Telephony Services architecture which provides a platform that enables the use of audio based communication services in Dwesa (Kunjuzwa et al., 2009);
- ❖ A Service for Personal Communication, Synchronous and Asynchronous, which is a one-stop shop that provides web services such as Electronic Mail, Short Message Services (SMS) and Multimedia Message Services (MMS) to foster communication in Dwesa (Samalenge et al., 2010);
- ❖ A Revenue Management System to help Siyakhula living lab generate enough revenue to cover its operational expenses (Ngwenya et al., 2010); and
- ❖ A middleware platform (Teleweaver) which provides an environment in which applications could interconnect and interoperate (Moyo et al., 2010).

This project looks, specifically, at the provision of a Text–To–Speech platform to assist people, living in marginalized areas of Eastern Cape, who are not able to read and write isiXhosa language.

1.4 Research Problem

The use of ICT4D projects in marginalized communities has opened various avenues for socio-

economic development. The majority of communities with such initiatives have witnessed promising results in that they are able to converse with the rest of the world through the use of ICTs. However, there are some barriers to reaching the information society goals for some of the indigenous targeted populace; the most obvious is the language barrier. The assumption is that once ICTs have been deployed in the targeted communities, people should be able to consume the services that come with ICTs. However, in practice, only the fortunate few are able to do so. This is because ICT platforms and solutions are primarily utilized in English, but very few people in Dwesa use this language to communicate with the rest of the world. As mentioned in Section 1.3, there are some projects which are already deployed in the Dwesa area, however, these are all written in English. However, the majority of the people in marginalized areas use their mother tongue (in this case isiXhosa) to communicate. Some people cannot read and write isiXhosa but still want to partake in the use of ICT information. That being said, this project seeks to develop and implement a Text-To-Speech platform that can help people who cannot read and write isiXhosa, but who are able to speak it. The system seeks to improve ICT4D e-services' usability through the development of an isiXhosa Text-To-Speech system.

1.5 Research Question

This research seeks to answer the question – *“Can ICT4D eServices' usability be improved through the development of an isiXhosa text-to-speech system/module?”* In order to answer this question, the following sub-questions (from Section 1.3) will be addressed:

1. How do people interact with ICT tools in marginalized communities?

This question seeks to elicit some of the preferred methods used by people living in rural communities in which ICTs have been deployed. The results of this will help validate the aim of deploying the Text-To-Speech application to such areas.

2. How do illiterate people engage in communication?

These question measures some of the alternative ways by which illiterate people become involved with ICT tools. The results of this analysis will further assist in strengthening the feasibility of the implementation of the Text-To-Speech program.

3. *How does illiteracy affect the use of ICT tools?*

ICTs change peoples' lives, some prefer to adapt to new forms of communication such as social networks while others prefer their conventional means of communication. It has been observed that some people fail to adapt to new technology due to a language barrier or illiteracy issues while the literate and active ones adapt easily. This question seeks to investigate various ways by which illiterate people are affected in the use of ICT tools.

4. *To what extent is language a barrier to ICT use?*

This question seeks to address the issue of a language barrier in a marginalized area, which affects most old people.

5. *Does isiXhosa Text-To-Speech (TTS) system improve the usability and uptake of ICT tools/services?*

The ICT projects currently deployed in marginalized rural areas are written in English, which is the problem for people who use isiXhosa as their language of communication. Therefore, not all people are able to use the ICT tools because of the language and they feel like they are not accommodated in this project. The Text-to-Speech system for the isiXhosa language is developed to improve the usability of ICT tools because it accommodates every individual in the marginalized areas under discussion. This question seeks to investigate the importance of a Text-to-Speech system in improving the usability of ICT tools.

6. *What is the most preferred method of communication employed by rural communities?*

This question seeks to identify the methods by which people prefer to interact with ICT tools. The results will help to ascertain which method can be used to improve communication between the ICT tools and community members.

7. *How to design and implement a software package for converting isiXhosa text to isiXhosa speech through prototyping and implementation?*

This question seeks to check whether it is possible to implement a software package which can convert text into speech and be used in rural areas. The results will show the importance of the Text-to-Speech system in rural areas.

1.6 Objectives of the Research

The main objective of the study is to improve the usability of Information and Communication Technology for Development (ICT4D) eServices through the development of the isiXhosa Text-To-Speech (TTS) system/modules.

The sub-objectives are:

- ❖ To investigate the most preferred method of communication employed by rural communities with ICT tools, by conducting a literature review and through observation.
- ❖ To investigate the ways in which people interact with ICT tools in marginalized communities, by performing a literature review and holding interviews.
- ❖ To assess the ways in which illiterate people engage in communication, by conducting literature surveys and through observation.
- ❖ To investigate the extent to which language is a barrier to ICT use, by conducting informal interviews, performing a literature review and through observation.
- ❖ To assess the ways in which illiterate people affect the use of ICT tools, by performing a literature review, conducting informal interviews, distributing questionnaires and through observation.
- ❖ To assess ways in which the isiXhosa TTS system improves the usability and uptake of ICT tools/services, by conducting informal interviews, performing a literature review and through observation.

- ❖ To design and implement a software package for converting isiXhosa text into isiXhosa speech through prototyping and implementation.

The questions, objectives and methodologies that used to address these sub-objectives are presented in Table 1-1 as follows.

Table 1-1: Summary of Questions, Objectives and Methodology

Questions	Objectives	Methodology
1. How do people interact with ICT tools in marginalized communities?	To investigate the ways in which people interact with ICT tools in marginalized communities.	-Literature review -Interviews
2. How do illiterate people engage in communication?	To assess the ways in which illiterate people engage in communication.	-Literature review -Observation
3. How does illiteracy affect the use of ICT tools?	To assess ways in which illiteracy affects the use of ICT tools.	-Literature review -Interviews -Questionnaire -Observation
4. To what extent is language a barrier to ICT use?	To investigate the extent to which language is a barrier to ICT use.	-Literature review -Interviews -Observation
5. Does the isiXhosa TTS system improve the usability and uptake of ICT tools/services?	To assess the ways in which the isiXhosa TTS system improves the usability and uptake of ICT tools/services.	-Literature review -Interviews -Observation
6. What is the most preferred method of communication employed by rural communities?	To investigate what the most preferred method of communication employed by rural communities is.	-Literature review -Observation
7. How to design and implement a software package for converting isiXhosa text to isiXhosa speech through prototyping and implementation?	To design and implement a software package for converting isiXhosa text into isiXhosa speech through prototyping and implementation.	-Literature review -Implementation

1.7 Research Methodology

This research employed a mixture of methodologies which includes: literature review, observations, interviews, implementation, testing and evaluation as discussed below:

- ❖ **Literature review:** Conducting a literature survey helped the researcher to understand various related projects in order to determine the feasibility of the implementation of the envisaged project. This included all information on the use of Text-to-Speech programs, which gave the researcher an initial understanding of the requirements. In all, the literature review provided some background information regarding this area of study. The ultimate goal of the literature review, in this project, was to furnish the researcher with current information on topics and other research which has been undertaken in the area of Text-To-Speech technology. We notice that there is a lot of work which has already been undertaken in this field. To engage in a fruitful literature review, the following sources of information will be used: journal publications, text books and conference proceedings.
- ❖ **Observation:** This methodology was employed to investigate some trends in the use of ICT tools in Dwesa. The method was used to investigate whether the Dwesa area contains all the characteristics of the rural areas of Eastern Cape. We observed that the area does indeed contain all the characteristics, because there is a lack of transport and poor infrastructure. During our visits to Dwesa we observed that there are same activities being performed, namely, the practice of isiXhosa cultural activities. The other observation was that community members were not aware of the Text-to-Speech system and its functionalities.
- ❖ **Interviews:** During regular visits to Dwesa, to conduct research on this field of study, informal interviews were conducted amongst the students, teachers and other community members, so as to elicit the system requirements based on users' perspectives. Interviews were meant to investigate the level of literacy in the area which would be used to support our proposed Text-to-Speech program. Interviews were also meant to investigate the basic understanding of users regarding computers and the Text-to-Speech system that is to be introduced to them.
- ❖ **Implementation:** The end results of the research implementation was to develop software

that is able to convert isiXhosa text into isiXhosa speech, that could be used in rural areas of the Eastern Cape.

- ❖ **Testing:** Testing the designed and implemented system was also part of the methodology related to the implementation of the Text-to-Speech system. Testing of the system was based on the functionality and usability approach. The testing was done after the system has been fully developed; some of the things that were tested included the user friendliness and naturalness of the system.
- ❖ **Evaluation:** As part of checking whether the implemented system reached the expectations of the intended users, a method called evaluation was considered. This was necessary to determine what the system has achieved and what it has left out, after the testing was conducted. Part of the expectations from the evaluation was to make sure that the citizens are fully satisfied and that they experience no major struggles in using the system. The satisfaction of users was very important so that they can use the system effectively to solve the problem of a language barrier in marginalized areas.

1.8 Motivation of Research

As mentioned in Section 1.3, there are some projects which are already deployed in the Dwesa community, but the problem is that they are written in English. Since the majority of the people who live in those marginalized areas are illiterate, since they cannot read and write a simple isiXhosa text, it is difficult for them to benefit from the projects written in English. The main motivation of the implementation of this system is to accommodate such people, who are willing to use ICT tools. There is a communication barrier between members of the Dwesa community, primarily due to literacy disparities. Hence, we propose a system to synchronize the interfaces of all the projects deployed through the use of the Text-To-Speech system.

1.9 Contribution of the Thesis

This thesis has created unity amongst the community members, especially between those who can write and read isiXhosa and those who cannot, because it is written in the language which is being used in the rural areas of the Eastern Cape. The other contribution that was made by the thesis was that it improves the use of computers as community members are able to make use of

the system. The main contribution made by this thesis is the development of intelligent Text-To-Speech conversion software. The unique feature of this thesis is that the system will be used in rural areas so as to improve the usability of ICT tools.

1.10 Structure of the thesis

The rest of this thesis is organized as follows:

In Chapter 2, a literature review of related work is provided. In Chapter 3, the system design and architecture are presented. The implementation of the system is outline in Chapter 4. In Chapter 5, the results, after conducting the functional and non-functional testing of the system, is presented. The study is concluded in Chapter 6.

1.11 Conclusion

This chapter has offered an introduction to the study. The problems and challenges that are facing rural community of Dwesa have been identified here. The implementation of the Text-To-Speech system for rural areas is a solution to the problems and challenges that people face concerning the issue of communication amongst community members. The research context and background information to the study has been outlined. A list of objectives to be achieved is given with the associated methods to be followed in order to achieve them. The next chapter provides a detailed literature review of the research area.

2 CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This Chapter presents the background of the research and offer a review of the literature in relation to the Text-To-Speech system. This Chapter begins with Section 2.2, in which an overview of the Text-to-Speech system is provided. In Section 2.3, the Text-to-Speech synthesis is presented, In Section 2.4, the Text-to-Speech architecture is presented. In Section 2.5, Text-To-Speech engines are presented and in Section 2.6, the application of speech synthesis is discussed. In Section 2.7, the topic of ICT in marginalized rural areas is covered. In Section 2.8, ICT4D in developing countries is explored, while in Section 2.9, the usability of ICT tools in rural areas is discussed. In Section 2.10, the study area, which is Dwesa in this case is presented, in Section 2.11, an overview of isiXhosa is provided. In Section 2.12, the development of isiXhosa Text-to-Speech is presented and, in Section 2.13, the chapter is concluded with the provision of details regarding related work in the field of Text-to-Speech. .

2.2 Overview of Text-to-Speech

The first Text-to-Speech system was built by Christian Kratzenburg in 1779 (Rousseau et al., 2004). The system was able to produce five long vowels such as the a, e, i, o, u sounds using some applications, called resonators activated. This system led to numerous Text-to-Speech applications that are being used in different applications today. There are some techniques that are being used in the implementation of the system, in order to produce a natural sounding voice. There are three main techniques that were used to implement this system, these are: articulatory synthesis, formant synthesis and concatenative synthesis (Rousseau et al., 2004).

The articulatory synthesis technique uses the human articulators such as the tongue and teeth to speak and vocal cords. The most important thing about this method is that it allows the changes to occur in the vocal tract so that it can produce an accurate utterance (Rousseau et al., 2004). The disadvantage of articulatory synthesis is that it is very difficult to implement and is very rarely used in practice.

Formant synthesis was used to model the formant frequencies of human speech. The advantage of formant synthesis is that it can contain a number of sounds, which makes it more flexible. It is easy to integrate with other methods of speech synthesis. This method is easy to adjust and it can produce a high quality of speech, if it is well adjusted, but it is still difficult to get full natural sounding speech.

The concatenative synthesis is the most important technique that can be used to produce more natural sounding speech. The technique uses a pre-recorded voice from a language that is being implemented (Rousseau et al., 2004). There are two main types of Concatenative synthesis that can be used to produce a naturally sounding speech. The unit selection synthesis is one of the types. The unit selection synthesis is uses a large databases of recorded speech information. When the database is being created each recorded utterance is segmented into all the following phones: individual phones, diphones, half-phones, syllables, morphemes, words, phrases and sentences. The units in the speech database is then created based on the segmentation and acoustic parameters such as the fundamental frequency (pitch), duration, position in the syllable and neighboring phones. This type of Concatenative synthesis provides the greatest naturalness of speech because it only apply a small amount of digital signal processing (DSP) to the recorded speech (Chauhan, 2011). Another type is diphone synthesis which uses a minimal speech database containing all the diphones occurring in a language. In diphone synthesis only one example of each diphone is contained in the speech database. This two types of Concatenative synthesis are used to make the speech sound more natural.

All these speech techniques were explored since the research was trying to build a machine that would be able to create human speech. Electronic signal processing was invented. The main idea was that the system should generate speech that is similar to that of a human being. However, a this machine came into existence in the last 50 years, before we actually saw what could be termed a practical example of the Text-to-Speech system. There were so many examples of this system that researchers tried to design. It is probably that the first practical application of speech synthesis was in 1936 when the United Kingdom (UK) Telephone Company introduced a speaking clock. It used optical storage for the phrases, words and part-words which were appropriately concatenated to form complete sentences (Black, 2000).

Therefore, after 1936, Homer Dudley at Bell labs (Juang, 2004), developed a mechanical device that worked using pedals and machine keys. When you are moving the pedals they cause a sound like human speech. The machine was called VODER (Voice Operating Demonstrator), which worked like an organ to generate almost all recognizable speech. The VODER application was demonstrated in New York and San Francisco to test whether it can produce a sound like that of a human being. Much of the work in this field was primarily concerned with constructing the signal rather than generating the phones from some higher form, like text (Black, 2000).

In the early 70s the standard UNIX manual included commands to process the Text-to-Speech system, from text analysis, prosodic, phoneme generation and waveform synthesis through a specialized piece of hardware. UNIX had only about 16 installations at the time and most were located in Bell Labs at Murray Hill. Techniques were being developed to compress speech in a way that it could be more easily used in applications. The Texas Instruments Speak 'n Spell toy, released in the late 70s was one of the earlier examples of mass production of speech synthesis. The quality was poor but, at the time, it was very impressive (Black, 2000). Speech synthesis was basically encoded using LPC (Linear Predictive Coding) and it used, primarily, isolated words and letters although there were also a few phrases formed by concatenation. The simple Text-To-Speech system, based on specialized chips became popular on home computers, such as the BBC (British Broadcasting Corporation) Micro in the UK and Apple. It was later developed into a product of DECTalk, which produces a somewhat robotic but very understandable form of speech. Before 1980, speech synthesis research was limited to large laboratories that could afford to invest time and money for hardware. By the mid-80s more labs and universities started to join in as the hardware costs dropped (Black, 2000). By the late eighties purely software synthesizers became feasible; they not only produced reasonable quality speech but could also do so in near real time.

People began looking to the faster machine and large disk space to look to improving synthesis by using larger and more varied inventories for concatenative speech. Yoshinori Sagisaka at ATR in Japan developed nuu-talk nuutalk92 in the late 80s early 90s (Black, 2000); this used much larger inventories of concatenative units. Thus, instead of one example of each diphone unit there could be many and an automatic acoustic based selection was used to find the best selection of sub-word units from a fairly general database of speech (Black, 2000). However, in

spite of very high quality synthesis examples of it working, generalized unit selection also produced some very bad quality synthesis. The development of speech synthesis is not in isolation from other developments in speech technology.

There are now many more people who have the computational resources and interest in running speech applications. The ability to run such applications puts a demand on the technology to deliver both working recognition and acceptable quality speech synthesis (Black, 2000). The availability of semi-free and free synthesis systems, such as the MBROLA projects and the Festival Speech Synthesis System, makes the cost of entering the field of speech synthesis much lower and many more groups have now joined in the development. However, although we are now at the stage of talking computers, there is still much to do. We can now build synthesizers, in any language, that produce recognizable speech. However, if we are to use speech to receive information as easily as we do from humans, there is still much to do. Synthesized speech must be natural, controllable and efficient both in the rendering and in the building of new voices (Black, 2000).

Text-To-Speech systems have an enormous range of applications. Their first real use was in reading systems for the blind, where a system would read some text from a book and convert it into speech. These early systems of course sounded very mechanical, but their adoption by blind people was hardly surprising as the other options, of reading or having a real person do the reading, were often not possible. Today, quite a number of systems that facilitate human computer interaction for the blind exist, in which the TTS can help the user, navigate around a windows system. Apart from users who have little choice, as in the case of blind people, people's reactions to old style TTS are not particularly positive (Black, 2000).

While people may be somewhat impressed and quite happy to listen to a few sentences, the novelty of this soon wears off, in general. In recent years, the considerable advances in quality have changed the situation to such an extent that TTS systems are more common in a number of applications. The main use of TTS today is probably in call-centre automation, where a user calls to pay an electricity bill or book some travel tickets and conducts the entire transaction through an automatic dialogue system. Beyond this, TTS systems have been used for reading news stories, weather reports, travel directions and a wide variety of other applications (Black, 2000).

Other than the engineering aspects of Text-To-Speech system, it is worth commenting that research in this field has made a significant contribution to our general understanding of language. This has often been in the form of negative evidence, meaning that when a theory thought to be true was implemented in a TTS system it was shown to be false; in fact, as we shall see, many linguistic theories have fallen when tested in speech systems. TTS systems have made a good testing ground for many models and theories (Black, 2000). These clearly show that the Text-to-Speech system was and is very important for the communication and interaction of the human beings and machines.

2.3 Text To Speech Synthesis

The goal of Text-To-Speech (TTS) synthesis is to convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person or enable the automatic conversion of a sequence of type written words into their spoken form (Bickley et al., 1998). According to Rousseau (2004), a Text-To-Speech system, in the simplest words, is the conversion of text to a speech output using a computerized system. It therefore allows for communication between humans and machines through synthetic speech (Rousseau et al., 2004). Currently, many speech synthesis systems are available in most major languages such as English, Japanese, French, Spanish, Russian, Italian, Marathi, Telugu, Czech, Finnish, Hindi and successful results are obtained in various application areas (Black et al., 1998). However, thousands of the world's minor languages, such as isiXhosa, lack such technology and researchers in the area are scarce. According to Takara (2006), Text-To-Speech synthesis is a process which artificially produces synthetic speech for various applications such as services over the telephone, e-document reading and a speaking system for handicapped people (Takara et al., 2006). The methodology used in the Text-To-Speech system is to exploit acoustic representations of speech for synthesis, together with the linguistic analysis of text to extract correct pronunciations (context, what is being said), and prosody in context (melody of a sentence, how it is being said) (Schroeter, 1996).

Synthesis systems are commonly evaluated in terms of three characteristics: Accuracy of rendering the input text (does the text to speech system pronounce, e.g., acronyms, names, URLs, email addresses and knowledgeable human would?), intelligibility of the resulting voice

message (measured as a percentage of a test set that is understood), and perceived naturalness of the resulting speech (does Text-To-Speech sound like a recording of a live human?) (Schroeter, 1996). The Text-to-Speech system is represented in the architecture that shows how the components of the system interact with each other in order to produce natural sounding speech.

2.4 Text-to-Speech Architecture

The Text-to-Speech system contains different components that are used in the process of converting the text into a speech. The Text-to-Speech system is divided into two phases: first of all, the text goes through the analysis where the unnecessary information is removed and then the final information is used to generate the speech signal. The process of text conversion also contains phonetic analysis; this is when graphemes are converted into phonemes of the language that is being developed, the language in this case is isiXhosa.

2.4.1 Text

The raw text entered into a system uses any device such as a desktop or laptop computer. The text can be words, sentences, numbers and abbreviations which pass through the text analysis process. In this case, the isiXhosa words and sentences were used as input text using a desktop computer. The text was entered using the command line on the Linux operating system. Before it was entered there are a lot of commands that need to be followed in order for a text to be rendered as speech at the end of the process.

2.4.2 Text Analysis

Text Analysis is used to convert input text into sound form. The text analysis contains different processes such as text normalization which used to normalize the text so that the numbers and symbols become words; abbreviations are replaced by the corresponding words (Vila et al., 2009). The most challenging process in text analysis is the linguistic analysis because a computer cannot understand the text as humans do. Linguistic analysis depends on the language that being developed. Text analysis is very important because the pronunciation of word and its meaning depends on this process. This process was used to analyse the type of input and the natural sound of the speech depends on this process. After the text is being analyzed it passes through the

phonetic analysis.

2.4.3 Phonetic Analysis

Phonetic analysis is used to convert symbols into phonological symbols using a phonetic alphabet such as that of the International Phonetic Association (IPA). This IPA contains the phonemic symbols which are related to pronunciation. The grapheme-to-phoneme conversion is defined in this stage in which graphemes are converted into phonemes (Yvon et al., 1998). Figure 2-1 shows the components of Text-to-Speech Architecture.

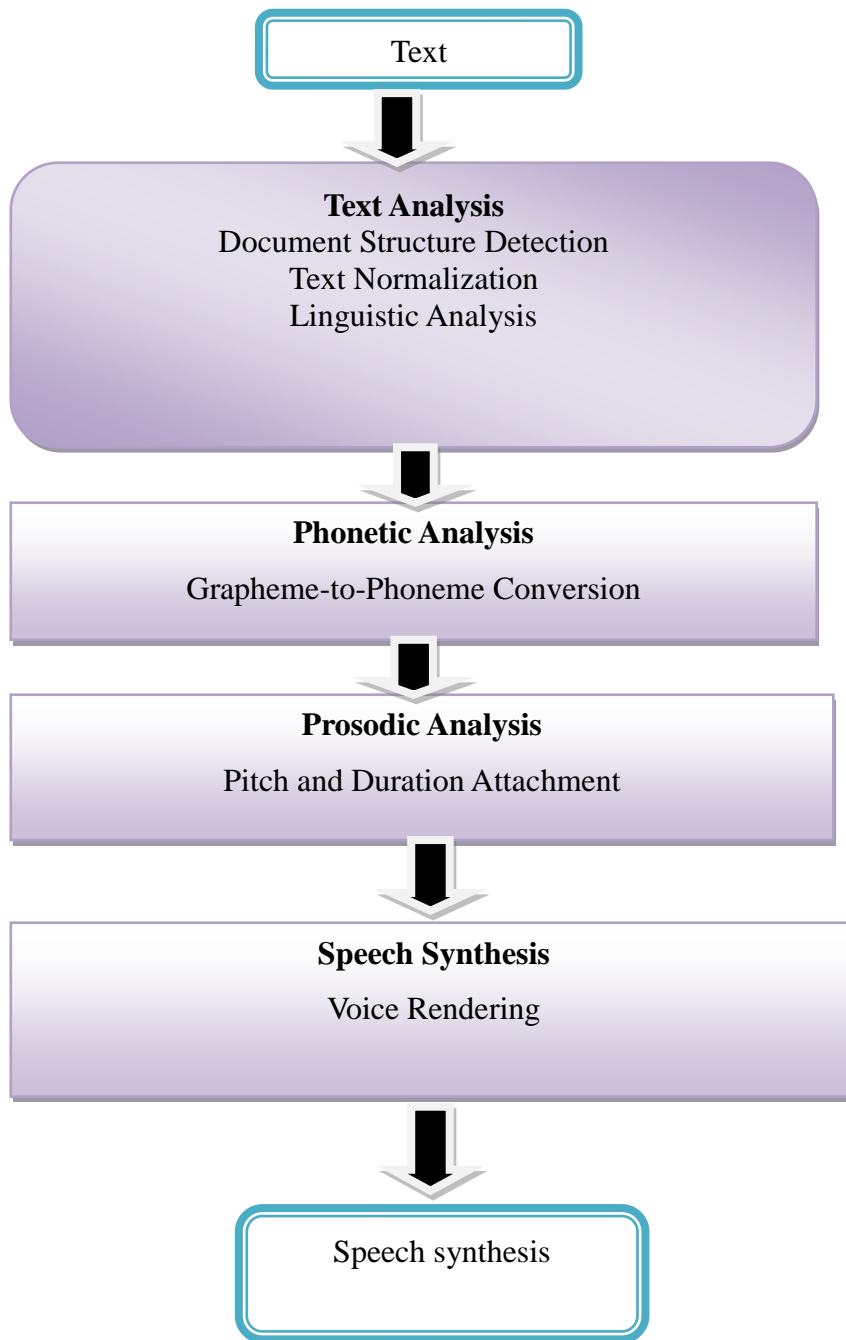


Figure 2-1: Text-to-Speech Architecture

2.4.4 Prosodic Analysis

Prosody is a concept which controls the naturalness of a speech by controlling stress patterns and the intonation process. This process plays a very important role in understanding speech; for instance, it checks and identifies the emotional state of a speaker and the background noise. Prosody is responsible for the natural sounding of the Text-to-Speech system by changing pitch and stress from the vowels and consonants.

2.4.5 Speech Synthesis

This is the final stage of the Text-to-Speech system which generates the speech signal. The raw text that was entered into a system in this stage is rendered as a speech. Figure 2.1 shows the complete stages of the Text-to-Speech system.

The components in Figure 2-1 show how the information is processed from a text, as input, into a speech synthesis (waveform), as output. The components of the Text-to-Speech system were implemented using different engines of speech.

2.5 Text-to-Speech Engines

In this Section, we discuss the different types of speech synthesis engines.

2.5.1 Festival Speech Engine

The Festival speech synthesis system is free open-source software for multi-lingual speech synthesis that runs a multiple platform offering blackbox Text-To-Speech, as well as opens architecture for research in speech synthesis. It is designed as a component of a large speech technology system, hence we provide a number of APIs. Festival was developed under the Linux operating system because the authors were more familiar with the operating system and it was also ported in Windows and is used in a number of real applications on that platform. Festival offers a general framework for building speech synthesis and it includes examples of various modules. Festival is great multi-lingual speech synthesis software available for Linux. It offers full Text-To-Speech conversion. It is a general multi-lingual speech synthesis system developed at the Centre for Speech Technology Research (CSTR) at the University of Edinburgh

(Black et al., 1998).

2.5.2 Flite Speech Engine

Flite is a small, fast, portable run-time synthesis engine; it was developed at Carnegie Mellon University (CMU); it is primarily designed for small embedded machines and a large server. Flite is designed as an alternative engine to Festival for various voices built using the Festvox suite of voice building tools. It is also designed as an alternative run-time synthesis platform for festival in applications where speed and size are important. It is written entirely in C language. Flite is compatible with festival speech synthesis and is not a replacement but a companion. It gives the same quality of voice as Festival but uses a small database and the voice gets compiled into a static structure. This type of engine uses no scheme, no interpreter, and no gc compiler, like Festival. The difference between Flite and Festival speech synthesis is that Festival is too slow, too big and lacks portability, when compared to Flite speech engine. Flite links voice with each utterance, voice is global (constant) and voice is linked to each synthesis function (e.g., appropriate text analysis, F0 model functions and appropriate models, and duration cart). Voice built using Festvox process may be compiled into efficient representation that can be linked against Flite to produce complete Text-To-Speech synthesizers. The system is free software (Black, 2000).

2.5.3 eSpeak Speech Engine

eSpeak is a Graphical User Interface (GUI) program which is used to prepare and compile phoneme data. eSpeak uses a formant synthesis method to create a voice. This allows many languages to be available in a small size. The speech is clear and can be used at high speeds, but is not as natural as a larger synthesizer, which is based on human speech recordings. eSpeak is available, as a command line (Linux and Windows), to speak text from a file and a shared library version for use by other programs.

eSpeak is regarded as one of the good speech synthesizers because it offers one a natural voice, but in a small size of a database. It also allows us to change the pitch and speed of the speech and support various languages and voices. eSpeak is a speech synthesizer for English (and several other languages) which will convert text to speech. It is executed immediately by using the

command line on an Ubuntu or a Linux operating system without any installation or configuration. eSpeak is open-source software which means that it is free. The Text-to-Speech system is used in different applications to meet the needs of the people in everyday life.

2.6 Applications of Speech Synthesis

Synthetic speech was used in many applications and it can be used in different applications. The system depends on what type of application you want to implement. Moreover, some applications, such as reading machines for the blind or electronic-mail readers, require unlimited vocabulary and a TTS system.

The application field of synthetic speech is expanding rapidly whilst the quality of TTS systems is also increasing steadily. Speech synthesis systems are also becoming more affordable for common customers, which makes these systems more suitable for everyday use (Klatt, 1987). For example, the increased availability of TTS systems may increase employment possibilities for people with communication difficulties. Speech synthesis has been applied in many different areas, as follows:

- ❖ **Application for the blind:** The Text-to-Speech system is very important for blind people. It is an application that is used to read text for the blind and for communication purposes. There is a command on Festival speech synthesis that can be used to read information from a book. This application can help blind people gain the necessary information. It was also used in school to assist blind learners because it is possible for them to read, but they can listen.
- ❖ **Education Application:** The Text-to-Speech system was used in many educational programs to teach students. Since computers already have this application, it can be used in the schools of blind and deaf people. It can be programmed for special tasks like teaching spelling and pronunciation in different languages. This application can allow even an educational program to be integrated on it. Text-To-Speech can close the gap between learners and teachers because some children may feel embarrassed to ask for help from teachers, especially deaf students (Klatt, 1987). The speech synthesizer, connected with a word processor, is also a helpful aid to proofreading.

- ❖ **Application for Telecommunication and Multimedia:** The newest applications in speech synthesis are in the area of multimedia. Synthesized speech has been used for decades in all kinds of telephone enquiry systems, but the quality has been far from good for common customers. Today, the quality has reached such a level that normal customers are adopting it for everyday use.
- ❖ **Illiterate people:** The Text-to-Speech system can be also used to help illiterate people, for communication purposes. In this case, illiterate people are categorized as people who cannot read and write their own language. These people are willing to partake in the Information and Communication Technology for Development projects in rural areas, but the language barrier is a problem. The Text-to-Speech system reads written text for illiterate people.

The fully interactive multimedia applications are automatic speech recognition systems, which are also needed. The automatic recognition of fluent speech is still far away, but the quality of current systems is, at least so good, that it can be used to give some control commands, such as yes/no, on/off, or ok/cancel (Klatt, 1987). However, all those applications were implemented for certain languages whilst a number of languages were not accommodated. For example, people who speak isiXhosa were not accommodated in these applications. The Text-To-Speech system that will be implemented during this study will help Xhosa people who have different challenges such as illiteracy, dumb and blind. The Text-to-Speech applications are also integrated into the Information and Communication Technologies currently being deployed for the marginalized rural areas of the Eastern Cape.

2.7 ICT in Marginalized Rural Areas

Information and Communications Technology (ICT) is an umbrella term that includes any communication device or application; it encompasses radio, television, cellular phones, computer and network hardware and software, satellite systems and so on. In addition, the various services and applications associated with them, such as videoconferencing and distance learning (Elisha, 2006) are also categorized under the term ICT. Bakar (2006) defines ICT as a diverse set of technological tools and resources used to communicate and to create, disseminate, store and

manage information.

According to COFISA (2008), ICT is a collective term referring to new and old technology that facilitates the processing and transfer of information across space and time (COFISA, 2008). The older communication technologies such as newspapers, radio and TV offer considerable unrealized potential. The new technologies, such as mobile phones and the internet, also have great potential to support the achievement of major development goals. These advantages include interactive forms of communication and low cost access to sources of life saving information (Curtain, 2003).

South African rural communities are still living under the subsistence level of having no access to the basic infrastructure essential for economic growth and development. This results in the movement of the youth from rural areas to urban areas, in search of employment opportunities (Acacia, 2000). Basic infrastructure, such as electricity, roads and telecommunication, is very important for economic growth in South Africa. Most of the schools in rural areas perform very badly because they have limited access to information and the use of the internet, which is considered vital in learning, training and business development in developing communities (Costello, 2000).

Most rural parts of South Africa lack any form of Information and Communication Technology (ICT); it is thus necessary to shape the South African information society by involving ICT skills that are required for economic growth (James, 2001). ICTs can be seen as catalysts in rural development as ICT projects enable the rapid development of rural areas. It is generally accepted that rural development is a multidimensional concept aimed at achieving the following (COFISA, 2008):

- ❖ Poverty alleviation in rural areas;
- ❖ Developing local economies in rural areas;
- ❖ Achieving basic standards of health, safety and other developmental infrastructure and services in rural areas;
- ❖ Encouraging and enabling rural people to invest in themselves and their communities;

- ❖ Cultural regeneration, including the development and integration of indigenous knowledge systems into a rural community's ways of doing things and learning;
- ❖ The long-term sustainability of livelihoods and improvements in quality of life.

In order to achieve rural development, ICTs have been deployed in these areas. The use of ICTs in enhancing and supporting rural development is highlighted by the emerging importance of knowledge as a key strategic resource for social and economic development (Pade, 2007). According to Furlonger (2002), urban scholars have many advantages when it comes to studies; they have access to computer centers, libraries and the internet where they can access information, experience teaching and they have an array of cultural facilities to choose from. Rural scholars have no textbooks, no electricity and desks, or even toilets. Herselman (2003) notes that 50% of school children in rural areas drop out before they reach high school, with little or no understanding of English. This observation alone shows that there are no experienced teachers because they are expected to know the language by grade five, at least.

According to Naidoo (2002), research has shown that the interval between rural and urban learners, in terms of basic skills such as reading and writing, is seven years. Information and Communication Technology (ICT) plays a leading role in the way in which information is distributed in South Africa and around the world. ICT aids teaching processes and in obtaining information in a dependent manner, between the researcher and the outside world, via communication and interaction.

The growth and development of Information and Communication Technologies (ICTs) has led to their wide diffusion and application, thus increasing their economic and social impact. The Organization for Economic Co-operation and Development (OECD) undertakes a wide range of activities aimed at improving our understanding of how ICTs contribute to sustainable economic growth and social well-being, and their role in the shift towards knowledge based societies. The phrase Information and Communication Technology refers to technology designed to access, process and transmit information. ICT for Development (ICT4D) refers to using ICT as tools to increase development effectiveness and efficiency, in marginalized rural areas.

ICT4D is based on three concrete notions: access (equal opportunities), networking (communication and organization); voice (participation in democratic process, good governance, cultural diversity and local content) (Weigal, 2004). According to Pade (2008), ICT plays a significant role in supporting rural development activities through providing supportive development information and creating essential interconnectivities between rural areas and more developed regions. However, in rural areas, ICT for Development (ICT4D) is still a working hypothesis faced with barriers and challenges associated with implementation and use in the rural environment this threatens the sustainability or relevance of the project or results in project failure. Thus, ICTs need commitment and support from different companies and organizations.

ICT projects require a lot of resources, including labour and finance. People who live in rural areas are struggling to access government services due to their remote locations. Most community members must travel long distances by foot and pay costly transport fees in order to access basic services. In their longitudinal household study of costs and coping strategies with chronic illnesses (Goudge et al., 2007), for example, they found that people in rural areas of South Africa do not go to free healthcare facilities because they cannot afford the transport.

2.8 ICT for Development (ICT4D) in Developing Countries

The Information and Communication Technology for Development (ICT4D) sector includes all projects that use ICT as a tool to attain development goals such as poverty alleviation or improved healthcare, environment and education for a society; all this falls under the umbrella of development. A common ICT4D project is one which seeks to bridge the digital divide and disparity between the haves and have nots in terms of ICT access and use. It is a relatively new field, and its projects are primarily rolled out in developing countries where ICT often marginalizes communities rather than strengthening them (Vosloo, 2003). Considering the diversity in origins, cultures, languages and beliefs within South Africa, ICTs in South Africa can offer a means of inclusion and integration (Consultants, 2008).

ICTs have the potential of transferring learning and an equally enabling environment within South Africa's dichotomous economy (Consultants, 2008). The South African government is actively involved in ICT projects throughout the country (Consultant, 2008). The government

has identified different departments and encourages all stakeholders who are willing to implement and support ICTs. The government, through several departments, carries out different duties in bringing ICTs close to everyone in marginalized areas of South Africa. All these departments work closely with each other and aim to support ICTs in South Africa and Africa as a whole to ensure rural development (Consultants, 2008).

According to the South African Cities Network, as of 2001, 42% of South Africans reside in rural areas (Statistics South Africa, 2006). This situation encouraged the development of infrastructure and services in rural areas (Jere, 2009). As government holds significant ownership within the telecommunications industry, it has been able to mobilize access and expand affordable access to rural regions; this includes a nearly 50% increase in households with telephones from 1996 to 2001 (Consultants, 2008). Therefore, people living in these communities need to have access to better services and, thus, the deployment of ICTs in their areas is vital. According to South Africa's ICT Development Framework, one of the goals for the Information and Communications Technology (ICT) sector in the country is to “increase the use of ICT as an enabler for socio-economic development, with equity”. In this regard, “the intent is to specifically address equity issues with regard to gender, disadvantaged groups and those in rural and under-served communities” (Afrika, 2007).

Table 2-1, below, shows some of the South African departments involved in the ICT and Development.

Table 2-1 South African Departments in ICT

Government Department	Activities
The Department of Science and Technology	Conducts research and development
The Department of Trade and Industry	Assists trade and business development of ICTs
The Departments of Labour	ICT skill and capacity development
The Department of Communication and partly the Department of Public Service, Administration and the Department of Public Enterprise	Oversees Telkom, South African Post Office, Sentech, South African Broadcasting Corporation, The National Electronic Media Institute of South Africa and the Independent Communications Authority of South Africa and provides access points for ICTs in rural areas

South Africa is leading in terms of Information and Communication Technology in Africa and has the most developed telecommunication network on the continent. Within the context of ICT4D, a number of e-services are developed and some of these include e-Commerce, e-Government, e-Health and e-Judiciary. Many other projects in ICT4D for marginalized areas have been deployed. All these projects are deployed under the supervision of the Siyakhula Living Lab in Dwesa. However, a problem persists because those who live in Dwesa are illiterate; they cannot write and read in the isiXhosa language, which is their mother tongue, whilst the projects are written in English. All the projects which are deployed in Dwesa are written in English, thus it is not easy for them to access the information. There exists a need to

develop a system which converts isiXhosa text into isiXhosa speech so that these people would be able to access the information. The Text-To-Speech system is currently under development for this language, for marginalized rural areas. The question is whether people are able to use the ICT4D projects which are currently deployed in marginalized rural areas and how often they communicate and interact with these applications. The most important reason for deploying such applications in rural areas was to improve the standard of living for community members. The usability of the system is very important in this case because the system is developed for a certain group of people.

2.9 ICT Usage in Marginalized Rural Areas

There are Information and Communication Technology (ICT) tools which are deployed in the rural areas of Dwesa in the Eastern Cape, within the context of the Siyakhula Living Lab. The main reason for deploying such tools was to improve the standard of living of the inhabitants. However, there is a problem concerning the ICT tools in the area, i.e. the problem of language. Language hinders the use of these tools because the tools are not written in the language of communication in the area; there is a language barrier amongst community members. The tools are written in English which is not the mother tongue of the Dwesa members. These ICT tools cause a digital divide amongst the community members, between those who can read and write English and those who cannot; other members feel like the ICT tools were deployed for educated people.

In the same area (Dwesa) there are some people who cannot read and write their mother tongue, which is isiXhosa; these people are considered to be illiterate. For these people it is far impossible for them to use ICT tools which are written in English while they cannot even read and write isiXhosa, the language they use to communicate. The problem of illiteracy hinders the use of ICT tools because, if the person is illiterate, it is not easy for them or they simply cannot use a computer. Even the instructions of how to use a computer are written in English. Illiterate people feel like they are not accommodated in the use of ICT tools because of the language.

The infrastructure is another thing which hinders the use of ICT tools in marginalized rural areas. In other areas of the Eastern Cape there is no electricity. The problem concerning infrastructure

is that in other areas there is no proper place to keep ICT equipment safe since there is no security, and no proper building.

The Text-to-Speech system will help people such as these because the illiterate cannot read and write isiXhosa, but they can listen and understand. The system converts isiXhosa text into isiXhosa speech. The Text-to-Speech system seeks to minimize the gap between community members. It also improves the usability of computers in marginalized rural areas because it draws all people to ICT tools and improves their skills. The Text-to-Speech system was developed to validate the used of ICT tools in marginalized rural areas. Dwesa is the area in which this system will be tested and deployed.

2.10 Study Area: Dwesa

Dwesa is a rural area located on the wild coast of the former homeland of Transkei, in the Eastern Cape province of South Africa. The community is under the Mbashe Municipality, which belongs to the Amatole region, based in East London. The nearest town is Willowvale, which is 50 km from Dwesa (Timmermans, 2004). In many ways, it is representative of many rural realities in South Africa and Africa as a whole. The Dwesa people are traditionally subsistence farmers who depend on the land for their livelihood (Palmer et al., 2002). The region features a coastal nature reserve which is an attraction, particularly to South African tourists; however they visit primarily during school holidays. The region has significant potential for both economic and cultural tourism due to the rich cultural heritage and the marine conservation project undertaken at the nature reserve. The nature reserve is a catalyst for tourism, which, together with government subsidies, is the main source of money for the local community. At the moment, the revenues are redirected to the administration/government offices in Bisho and do not benefit the community directly.

The only way in which tourism can benefit the community is by promoting local arts and crafts. Most residents of Dwesa depend on the state pension of the elderly people and other welfare payments. There are a number of activities ranging from basket-making to wood carving (Jere, 2009) which a number of community members partake in. Moreover, Dwesa is the site of conservation of the traditional culture of the Xhosa people and it has much to offer to the tourists

and to the outside world in terms of the preservation of traditional customs and ceremonies, dances and, especially, music. Unfortunately, like many other rural areas in South Africa or the Eastern Cape, Dwesa is characterized by a lack of infrastructure in terms of roads, electricity and widespread poverty, lack of services and isolation (Dalvit et al., 2007).

Isolation is the main reason for young people leaving Dwesa for urban areas, a typical phenomenon in rural areas (Salt, 1992). Government services (Department of Education and Department of Communication, 2001) have highlighted that information and communication technology (ICT) plays a key role in contributing to improving the situation of communities which are already disadvantaged in many other ways. There is a project in Dwesa called the Siyakhula Living Lab, which is meant to improve the quality of life in marginalized areas of South Africa. Information and Communication Technology skills are practiced in this project so as to teach those living in this area and to improve the performance of schools in rural areas. There are many projects that are already deployed in the Siyakhula Living Lab to help people living in rural communities to familiarize themselves with the use of technology (Dalvit et al., 2006).

The name Siyakhula means “we are growing together”. The Siyakhula Living Lab (SLL) was initiated by the Telkom Centre of Excellence at the University of Fort Hare and Rhodes University in 2005; both universities are located in the Eastern Cape Province of South Africa. The mission of the Siyakhula Living Lab is to develop the field-test prototype of a multi-functional, distributed community communication platform for deployment in marginalized and semi-marginalized communities in South Africa, where a large part of the South African population live (Siyakhula Living Lab, 2008).

Siyakhula Living Lab aims to develop the marginalized community by equipping people in the area with the necessary technological skills to support the projects which have been deployed. It shows how marginalized communities that are currently very difficult to reach may, in future, be joined with the greater south African and African communities to the economic, social and cultural benefit of all (Tarwireyi, 2008). The important features of the Siyakhula Living Lab are based on the close consultation and engagement of members of community. There is a wide range of disciplines involved as well as the close cooperation between these two Universities that

were separated for a long time during apartheid. From the University of Fort Hare and Rhodes University, the departments which are involved are: computer science, education, anthropology, information systems, communication and African languages, and more in future.

There are many projects that have already been deployed for the Dwesa community. Some of them are already running and some are still in the developmental process. The e-Commerce projects include e-Shopping, e-Health, e-Government, e-Judiciary, e-Mobile, Help Desk and several e-Services projects like novel interaction techniques for mobile phones, A middleware solution for a multi-purpose and a service for personal communication that are currently underway. All these projects are under the umbrella of the Siyakhula Living Lab. The projects are meant to improve the living standards of the Dwesa community (Siyakhula Living Lab, 2008). The SLL project has attracted several organizations in South Africa. SLL is currently sponsored by several organizations, such as:

- ❖ Telkom Centre of Excellence (COE) at the University of Fort Hare and Rhodes University,
- ❖ Technology and Human Resources for Industry Programme (THRIP) of the Department of Science and Technology of South Africa,
- ❖ Cooperating Framework on Innovation Systems between Finland and South Africa (COFISA), a programme of the South African and Finnish governments.
- ❖ A number of industry partners including Telkom SA, Tellabs, Amatole, Telecommunication, DRISA and Saab Grintek.

There are many other stakeholders and organizations that also play important roles in the success of the Siyakhula Living Lab. The developments of Information and Communication Technology (ICT) projects like in SLL have encouraged the use of technology in marginalized rural areas (Jere, 2009).

The driving force of the project is a group of young researchers from both universities and members of the projects. Members of the group pay regular monthly visits to Dwesa and stay there for approximately one week, each time. Siyakhula Living Lab sees that there is a need to introduce Information and Communication Technology (ICT) to marginalized communities (Siyakhula Living Lab, 2008). ICT projects improve the living standards of marginalized rural

areas of South Africa and Africa as whole. The Text-to-Speech system is developed for isiXhosa speakers who live in the rural areas of the Eastern Cape.

2.11 Overview of IsiXhosa

IsiXhosa is one of the eleven official languages of South Africa, which includes: English, Afrikaans, isiNdebele, Southern Sotho, isiZulu, Sepedi, Tshivenda, Setswana, siSwati and Xitsonga. IsiXhosa is one of the country's four Nguni languages; these include isiZulu, isiXhosa, siSwati and isiNdebele. The Nguni people are grouped into three different subgroups which are the Northern Nguni, the Southern Nguni and the Ndebele. The isiZulu and siSwati languages are classified as the Northern Nguni languages, while isiXhosa is classified as a Southern Nguni language. IsiXhosa is a tonal language, which is marked by a number of tongue-clicking sounds. The language (isiXhosa) is derived from the Koi-San language. IsiXhosa employs click sounds in the pronunciation of consonants, such as c, q, and x, which were most likely borrowed from the Koisian language as a result of long and extensive interaction between the Xhosa and Koisian people. The isiXhosa system was written with the Latin alphabet, which emerged from Christian missionaries during the early 19th century. Table 2.2 shows the Latin alphabet which is used in isiXhosa:

Table 2-2: isiXhosa pronunciation

A	b	Bh	d	Dl	dy	E	en	f	g	gr	h
Hl	i	J	k	Kh	kr	L	l	m	mbh	n	nd
Ndl	ndy	Ng	ng'	Nj	ntsh	Ng	o	p	ph	r	rh
Ndl	sh	T	th	Tl	tsh	Ty	u	v	w	y	z

Table 2-3: isiXhosa Click Consonants

C	Ch	Gc	Nc	ngc	nkc
Q	Qh	Gq	Nq	ngq	nkq
X	Xh	Gx	Nx	ngx	nkx

The isiXhosa alphabets are original from the Latin language. The isiXhosa have been grouped

with quite a few dialects which are: Xhosa (original), Gcaleka, Bhaca, Ngqika, Thembu, Mpondomise, Mfengu, Mpondo and Bomvana. All these groups of tribes have different rituals and/or ceremonies that they perform, such as circumcision.

According to South African statistics (Census, 2001), isiXhosa is the second most spoken language after isiZulu in the country, which is spoken (isiXhosa) by 17.6 % (percent) of all South Africans or 7907149 people. IsiXhosa is the regional language of most people living in the Eastern Cape; 83.4% of the populations of the Eastern Cape speak isiXhosa as their mother tongue. There are many Xhosa people who are living in other provinces such as the Western Cape, which consists of 13.6% of isiXhosa speakers. There are a few isiXhosa speakers in other provinces such as the Free State which consists of 9.1% Xhosa speakers, while the North West consists of 5.8% and Gauteng boasts a 7% of Xhosa population.

Fig 2.2 shows the map of isiXhosa speakers in the nine provinces of South Africa (South African statistics, 1996).

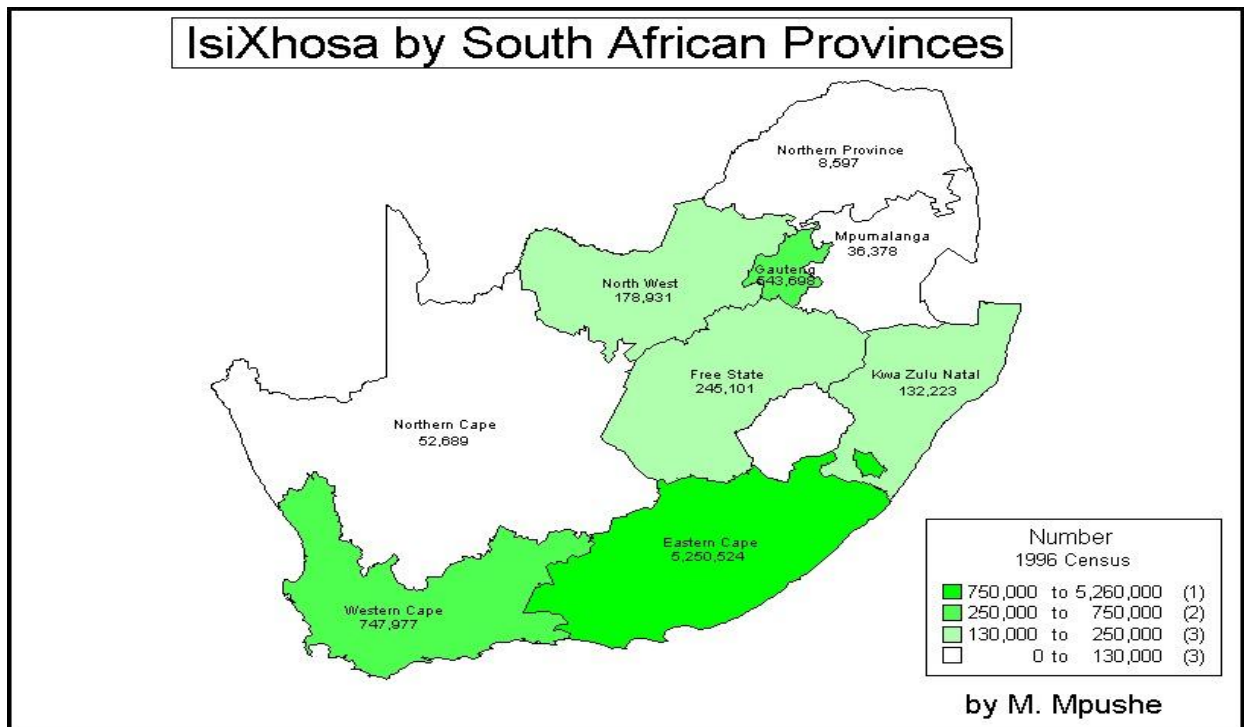


Figure 2-2: South African Provinces (Statistic South Africa, 2006)

According to Figure 2.2, the Eastern Cape Province has the highest frequency of isiXhosa speakers. Most Xhosa people live in rural areas, where there are no services such as electricity, roads and water supply. This work aims to implement an isiXhosa Text-To-Speech system that can be used by Xhosa people who live in the rural areas of South Africa. The system will be used by isiXhosa speakers.

2.12 Developing the IsiXhosa Text-to-Speech System

There is a lot of work that needs to be done in the field of speech technology, especially in South Africa. There are few languages that are already developed for Text-To-Speech synthesis in this country. The Text-To-Speech systems that have already been implemented, were used for commercial and academic purposes, while not a single one was implemented to accommodate the isiXhosa language. There is a need to implement an isiXhosa Text-To-Speech system so as to assist people who speak this language. Text-To-Speech systems were developed for different purposes, some to help blind people, some for educational purposes such as reading, learning a new language, some for deaf people to communicate with the hearing and some for telecommunication. The systems were designed with different programming languages such as C++ and Java. The concatenate module was used in quite a number of Text-To-Speech systems that were implemented for different languages. Concatenative is inexpensive and easy to update. Blind and deaf people who speak isiXhosa were not accommodated. The systems were implemented for developed urban areas, and not rural areas.

Most of the systems were developed to run in Windows. The difference between the systems that we are developing and those that already exist is that our system includes people who speak isiXhosa as their mother tongue. The system is developed for educational purposes. It is also developed for people who live in rural areas. We have used C++ and Festival schema to implement our system. The Linux operating system is used as a platform during the implementation of our system. There is a lot of work that is being done in the field of speech synthesis. There are other languages which are already developed in a Text-to-Speech system for different purposes.

2.13 Related Work

A lot of work has been done in this area of research. A Text-To-Speech engine has been used for developing Text-To-Speech in different languages. It was used in helping people learn new languages. There are many uses for Text-To-Speech systems, these include: digital actors, avatars, digitized speech (in conjunction with speech recognition and possibly a translator), as well as next generation chat applications. As such there is a lot of research in the area. The following are some of the better Text-To-Speech systems which were developed for commercial and academic purposes:

2.13.1 Commercial

First, the commercial speech synthesis system was mostly hardware based and the developing process was very time-consuming and expensive. Since computers have become more powerful, most synthesizers today are software based systems. Software based systems are easy to configure and update, and they are usually much less expensive than the hardware based systems. However, a stand-alone hardware device may still be the best solution when a portable system is needed. The Whistler Text-To-Speech was implemented.

Whistler Text-To-Speech (Windows Highly Intelligent Stochastic taLkER): was developed by Microsoft in 1998 and is, today, used in a number of their applications including Narrator as part of Windows 2000 and XP. The overall quality of the speech is not great and it sounds like the typical talking computer. The system is designed to produce synthetic speech that sounds natural and resembles the acoustic and prosodic characteristics of the original speaker; the results have been quite promising (Huang et al., 1996, Huang et al., 1997 and Acero, 1998). The speech engine is based on concatenative synthesis and the training procedure on Hidden Markov Models (HMM) (Hon et al., 1998). The project originally proposed a novel voice training system. Unfortunately, this research appears to have been discontinued and no information about the training was released. The present system has no facilities to make new voices (Hood, 2004).

Sanosse Text-To-Speech synthesis-: is another system that has been developed, originally for educational purposes for the University of Turku. The system is based on concatenative synthesis and is available for Windows 3.1/95/NT environments. The adjustable features are the speech

rate, word emphasis, and pauses between words. The input text can also be synthesized letter-by-letter, word-by-word, or even syllable-by-syllable. The feature can also be controlled with control characters within a text. Sanosse synthesis is currently used in aLexis software which is developed for computer based training for people with reading difficulties (Hakulinen, 1998). The original Sanosse system is also adopted by Sonera for their telephony applications.

SoftVoice Text-To-Speech-: developed by SoftVoice-Inc, was first written in 1979 and is now released as a set of Windows DLLs to enable programmers to easily add speech synthesis capabilities to their products. SoftVoice provides decent voice quality, but lacks any programs or documentation on how to create a new voice. The speech quality of SoftVoice is probably not the best of the available products but, with the large number of control characters and different voices, it is very useful for several kinds of multimedia applications (Hood, 2004).

HADIFIX Text-To-Speech (HALbsilben, Diphone, SuffIXE) -: is a TTS system for Germans, developed at the University of Bonn, Germany. The system is available for both male and female voices and supports control parameters, such as duration, pitch, word prominence and rhythm. The insertion of pauses and accent markers into the input text and the synthesis of the singing voice are also supported. The system is based on the concatenation of demisyllables, diphones, and suffixes (Portele et al., 1991, 1992). First, the input text is converted into phonemes with stress and phrasing information and then synthesized using different units. For example, the word *Strolch* is formed by concatenating *Stro* and *olch*.

The concatenation of two segments is done in three methods. Diphone concatenation is suitable when there is some kind of stable part between segments. Hard concatenation is the simplest case of putting samples together with, for example, glottal stops. This also happens at each syllable boundary in demisyllable systems. Soft concatenation takes place at the segment boundaries where the transitions must be smoothed by overlapping (Portele et al., 1994).

Lucent TTS engine-: was developed by Bell Laboratories. The engine is used in a number of Bells own telecommunication and messaging products and it is available to other developers. The system works primarily within the context of a telephone system and is obviously written with that in mind. Natural Voices is an ongoing research project at AT&T (Hood, 2004).

The aim of this project was to get as natural sounding speech as possible, for use in telephone systems. The quality of the output produced by the system is very impressive and has been used in a number of movies and recognized as a leading example by Discovery Magazine in their article, the Mathematics of Artificial Speech (Hood, 2004).

Laureate Text to Speech-: is a speech synthesis system developed during this decade at BT (British Telecom) Laboratories. To achieve good platform independence, Laureate is written in standard ANSI C (Gave, 1993 & Morton, 1987). The Laureate system is optimized for telephony applications so that a significant amount of attention is paid to the fields of text normalization and pronunciation. The system also supports multi-channel capabilities and other features needed in telecommunication applications. The current version of Laureate is only available in British and American English, with several different accents. Prototype versions for French and Spanish also exist and several other European languages are under development (Breen et al., 1996).

DECTalk, Digital Equipment Corporation (DEC) Text to Speech-: also has long traditions with speech synthesizers. The present system is available for American English, German and Spanish and offers nine different voice personalities: four male, four female and one child. The present system probably has one of the best designed text preprocessing and pronunciation controls. The system is capable of saying most proper names, e-mail and URL addresses and supports a customized pronunciation dictionary. It also has punctuation controls for pauses, pitch, and stress, and the voice control commands may be inserted in a text file for use by DECTalk software applications. In addition, the generation of single tones and DTMF (Dual Tone Multi Frequencies) signals for telephony applications is supported.

DECTalk software is currently available for Windows 95/NT environments and for Alpha systems running Windows NT or DIGITAL UNIX (Hallahan, 1996). The present DECTalk system is based on digital formant synthesis. The synthesizer input is derived from phonemic symbols instead of using stored formant patterns, as in a conventional formant synthesizer (Hallahan, 1996). The system uses 50 different phonemic symbols including consonants, vowels, diphthongs, allophones, and silence. Symbols are based on the Arpabet phoneme alphabet which is developed to represent American English phonemes with normal ASCII characters (Waters et al., 1993).

2.13.2 Academic

Black and Taylor developed a Text-To-Speech system at the Center for Speech Technology Research. It was used in reading systems for the blind; in these systems, the system would read some text from a book and convert it into speech. The current system is available for American and British English, Spanish, and Welsh. The system is written in C++ and supports residual excited LPC and PSOLA methods and MBROLA database. With the LPC method, the residuals and LPC coefficients are used as control parameters. As a university program, the system is available free-of-charge for educational, research, and individual use (Black et al., 1998).

The system is developed for three different aspects: namely, for those who want to simply use the system from arbitrary Text-to-Speech system for people who are developing language systems and wish to include synthesis output, such as different voices, specific phrasing, dialog types and so on, and for those who are developing and testing new synthesis methods (Black et al., 1998).

However, in this work we propose the use of isiXhosa language. The Xhosa people who are blind were not included in this study.

Alam developed a Text-To-Speech system at the BRAC University in Bangladesh for Bangla speakers. Besides the obvious uses of a Text-To-Speech (TTS) system, from listening to computerized books to one's email, it also allows the visually impaired and those who cannot read the Bangla language to access the Bangla information via electronic content such as the World Wide Web. Imagine a blind person being able to use a computer almost as efficiently as someone with eyes, all s/he does is to take the mouse cursor to a certain position in the screen and the computer reads the words to him aloud. The target of this system includes three kinds of people: illiterate people, the visually impaired and people who cannot read the Bangla language (Alam et al., 2007). However, the system was developed for Bangla speakers who have different speech and life orientation forms to South African people. IsiXhosa speakers experience the same problems as the Bangla people since they were not accommodated in the development of the system.

Gakuru, at the University of Nairobi in Kenya, developed a Text-To-Speech system for Swahili speakers. Swahili has now been recognized as a regional language in the East African region with almost 200 million speakers. The Swahili Text-To-Speech system was developed for market purposes for major companies such as Microsoft and Google. This fact has been recognized by major Information Technology (IT) companies such as Microsoft and Google whose products, Microsoft windows and the Google search engine, are already available in Swahili. Microsoft is also in the process of localizing its Microsoft office to Swahili (Gakuru, 2005).

The potential for Swahili as a market for IT products now appears to have been fully acknowledged. It was also used by a large number of blind children in schools to communicate (Gakuru et al., 2005).

However, all these systems were designed but not for isiXhosa speakers. The blind children who speak isiXhosa were not involved in these systems that were designed. The Swahili system was developed for market purposes; however, our system is free and educational.

Barnard developed a Text-To-Speech system for the isiZulu language. The development was aimed at fostering an understanding of the challenges involved in developing a Text-To-Speech engine for a Nguni language; it consisted of two phases. Initially, a fairly primitive synthesizer was developed using a limited set of sentences and an early version of components, such as a letter to sound converter, were used (Barnard et al., 2005).

Bosch developed a Text-To-Speech system in a human language technology research group within the Meraka Institute, specialists in developing open source software for language and speech technology applications inter alia, a Text-To-Speech system for isiZulu speakers. This is the first of a number of local languages for open-source TTS systems currently being developed (Bosch, 2009). However, the open-source software was not going to be used by isiXhosa speakers.

Mashao developed the Text-To-Speech synthesis of an Afrikaans language that was based on diphones. Using diphones makes the system flexible but presents other challenges. A previous effort to design an Afrikaans TTS system was made by SUN. They implemented a TTS system

based on full words. A full word based TTS system produces more natural sounding speech than when the system is designed using other techniques. The disadvantage of using full words is that it lacks flexibility. The baseline system was built using the Festival Speech Synthesis System. Problems occurred in the baseline due to the mislabeling of the diphone and the diphone index. The system was improved by manually labeling the diphone using Waves, and by changing the diphone index (Mashao et al., 2005).

Bali developed a Hindi Text-To-Speech system. Hindi is one of the official languages used in India; it is spoken as a first language by 33 percent (%) of the Indian population, and by many more as a lingua franca. However, good quality Hindi TTS systems that can be used for real-time deployment are not available. Though a number of research prototypes of Indian language TTS systems have been developed, none of these are of a quality that can be compared to commercial grade TTS systems in languages like English, German and French (Bali et al., 2004).

2.14 Conclusion

ICTs have expanded to most rural areas of South Africa. The use of ICTs in rural areas produces more skills amongst the members of rural areas, such as the use of computer with internet. The isiXhosa background was described in detail in this chapter. It also discussed the different applications for text-to-speech. The chapter also provides an overview of the Dwesa area and ICT projects. The ICT4D was also discussed. The chapter concludes by comparing Text-Tto-Speech engines. The ensuing chapter presents the design and architecture of the system.

3 CHAPTER 3: SYSTEM DESIGN AND ARCHITECTURE

3.1 Introduction

The system design is defined as a process of defining the architecture, components, modules, interface and the data required for a system to satisfy specified requirements (Ngwenya et al., 2010). The main purpose of a system design, in this research, is to create technical solutions that satisfy the functional requirements of the system (NYS Project Management Guide Book, 2010). This approach is the most interesting and most difficult to undertake and it is usually used, in engineering or computer science, to carry the entire design. In this chapter, the system back-end design is presented in Section 3.2. Section 3.3 presents the database design of Festival speech synthesis. In Section 3.4, the user's role in the system is presented and in Section 3.5, and outline of the system architecture is provided. In Section 3.6, the design of the system is presented and, in Section 3.7, the front-end design architecture is provided. In Section 3.8, the chapter is concluded with a presentation of the design considerations of the system.

3.2 System Back-end Design

The purpose of the system back-end is to support front-end services and allow users who interact with applications on the front-end system to make requests on the back-end system. The back-end design of the Text-To-Speech system is used to control the functionality and interaction of the system with the command line on the Linux operating system. This includes functionalities such as taking words or sentences from the database.

In addition to the back-end design, there is a MySQL database, which forms part of this system design. The connection and communication of the MySQL database, with the entire system, was done through the implementation of the Festival schema script. The MySQL database is free open-source software which is suitable for a system developed for marginalized rural areas. Figure 3.1, below, shows part of the back-end and part of a front-end system design.

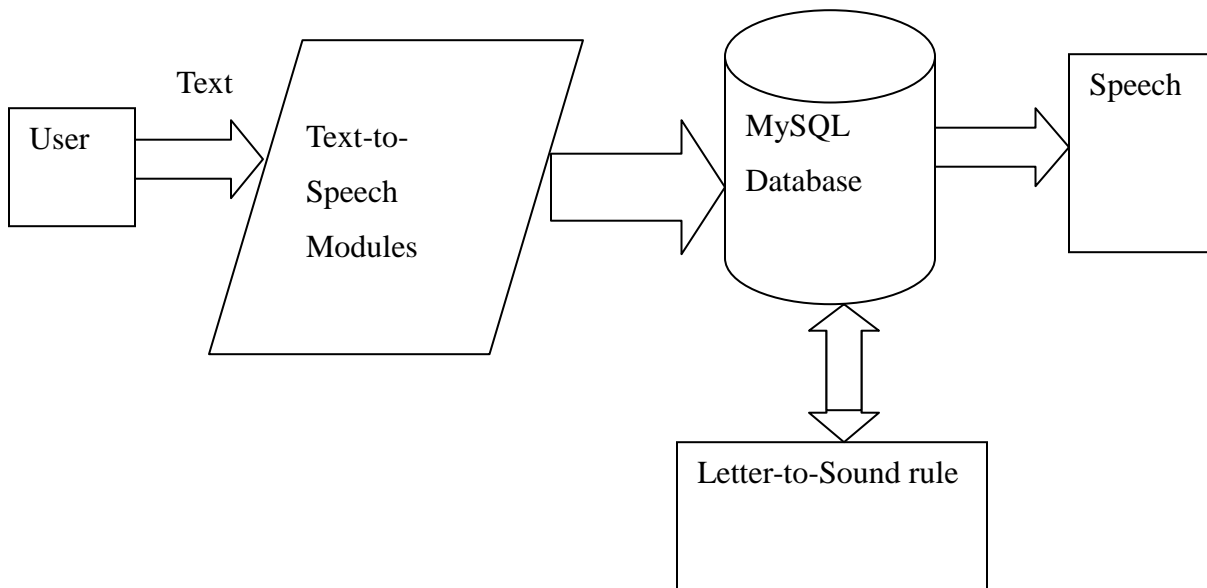


Figure 3-1: Front-end and Back-end Architecture

Figure 3-1, above, shows the interaction and communication of the Text-to-Speech modules and the MySQL database. Figure 3-1 depicts the back-end and front-end design of the system. As shown in Figure 3-1, the user type text (word or a sentence) using a terminal command line on the Linux operating system. Text passes through the Text-to-Speech modules. The Text-to-Speech requests a word or sentence from the MySQL database. If the word is not in the database MySQL sends a request to a Letter-to-Sound rule which is responsible for creating words which are in the database. After the Letter-to-Sound rule construct the word, the text is sent back to the MySQL and it is rendered as a speech.

3.3 Database Design on Festival

The isiXhosa database was designed so that it would be suitable for marginalized rural areas. The database design offers an idea of how the information flows in the database tables. The database design involves the use of data modeling which is commonly defined as an activity in the software development process of information system, which typically uses a data management system to store information (Merson, 2009). The output of this activity is the data model, which describes the static information structure in terms of data entities and their relationships. This structure is often represented graphically in entity-relationship diagram (ERD)

(Merson, 2009). The data model facilitates the communication and the connection between database tables. This process involves three important activities, namely: entity, attributes and relationship which are used to represent a data in table form. An entity is defined as a person, place, thing or event of interest to the organization (Balland, 1998). The attribute describes the characteristics of properties of the entities. And relationship is represented with lines between entities. It depicts the structural interaction and association among the entities in a model (Balland, 1998). Figure 3-2, below, shows the database model used to present the tables which contain information for isiXhosa.

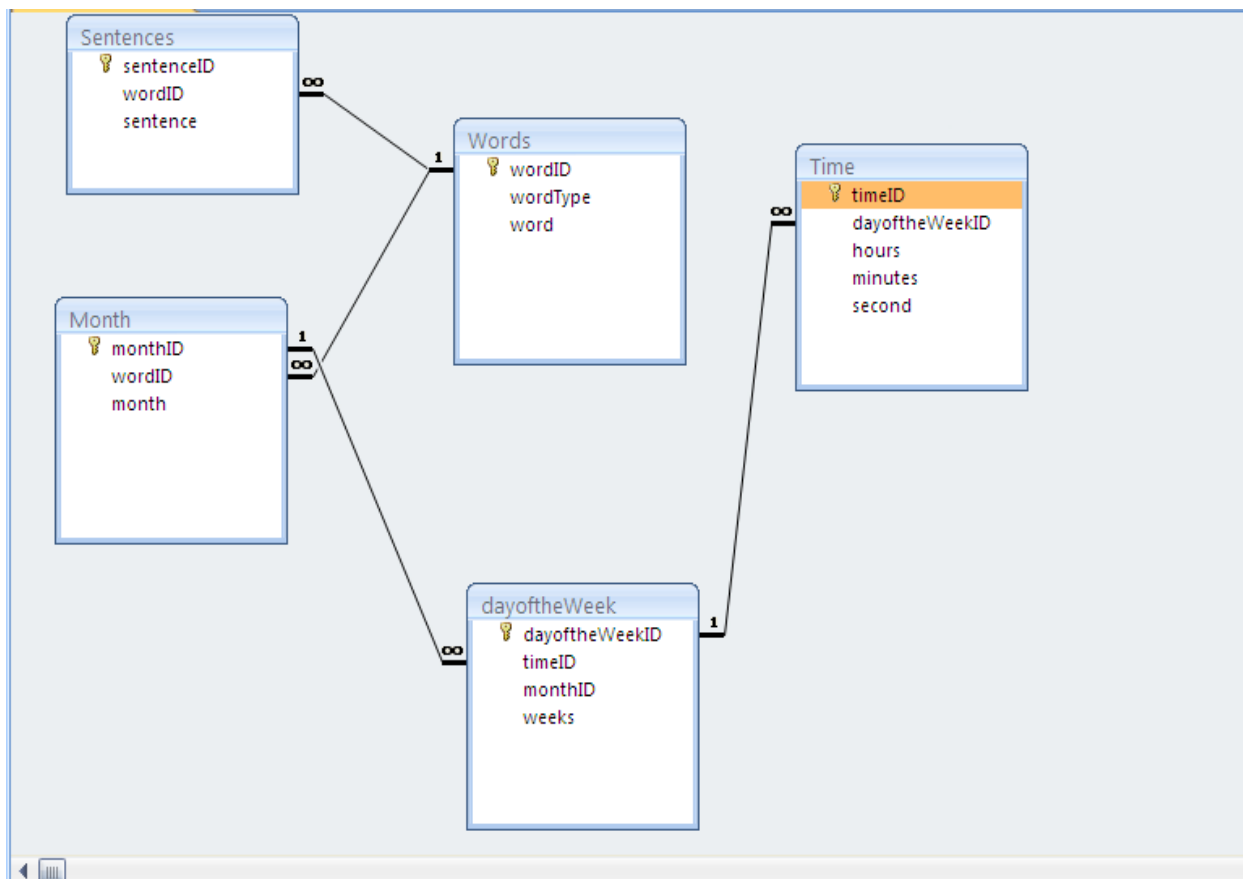


Figure 3-2: Database Model

Figure 3-2, present the Entity-Relationship Diagram (ERD) that involved the tables of the database that stored and capture isiXhosa information such as words and sentences. There is a relationship between tables that contains information of Text-to-Speech. The relationship between tables is One-to-Many (1-to-m). There is gold key next to each ID that represents the

primary key of table. There is a relationship between *words* table and *sentence* table. The relationship is that a sentence can be made up of words. These two tables connected with a secondary key from *words* table to *sentence* table. Words table also connected to month table because months can be made up of words. The relationship between month table and day of the week is that month can be made up of days of the week. During there is time that is involved, there is a relationship between time table and days of the week table. In Figure 3-2, all the tables have different primary key and secondary that show the relationship between tables.

3.3.1 User Roles

The Text-to-Speech system was designed and implemented with three kinds of users in mind. All the users have different functionalities in the system. These users include the *Administrator*, *Community members* and *Students*. These users have different authorities in the Text-to-Speech system. The following sections discuss the role of each user in the system.

3.3.2 Administrator

The administrator in the Text-to-Speech system is responsible for the maintenance of the overall tasks of the system. Figure 3-3 shows that the administrator has a lot of functions to perform, in comparison to other users. The administrator is responsible for adding words and sentences to the system database. The administrator can also manage how the users use system. Since the Text-to-Speech system was developed for illiterate people, i.e. those who cannot read and write in their mother tongue. The administrator is responsible for assisting the users on how to use the system. The administrator can also remove words or sentences which are not necessary. For instance, isiXhosa has some borrowed words from English. The administrator is responsible for changing or removing such words from the database.

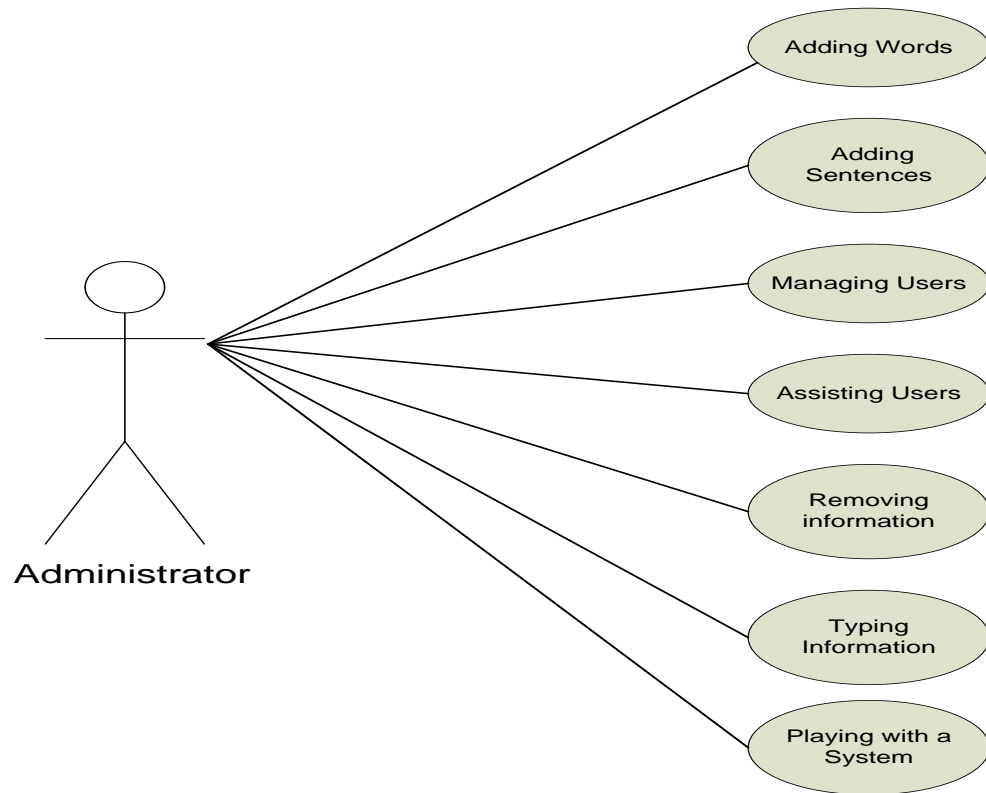


Figure 3-3: Administrator Use Case Diagram

3.3.3 Community Members

Community members have no authority to add or change anything from the database. The authority was given only to the administrator in order for him/her to maintain the system. The community has little functionality that is being performed in the system, their rights are limited. They are responsible for interacting with the system in the front-end design without knowing what happens at the back-end design. Community members are responsible for the information or text they type in a command line requesting the system to convert the text-to-a speech. They can familiarize themselves by playing with the system, in doing so they are also improving their skills in the use of ICT tools; this is one of the research objectives. Community members can assist each other in the absence of an administrator, if some already know how to interact with the system. Figure 3-3, shows the user case diagram of the community members.

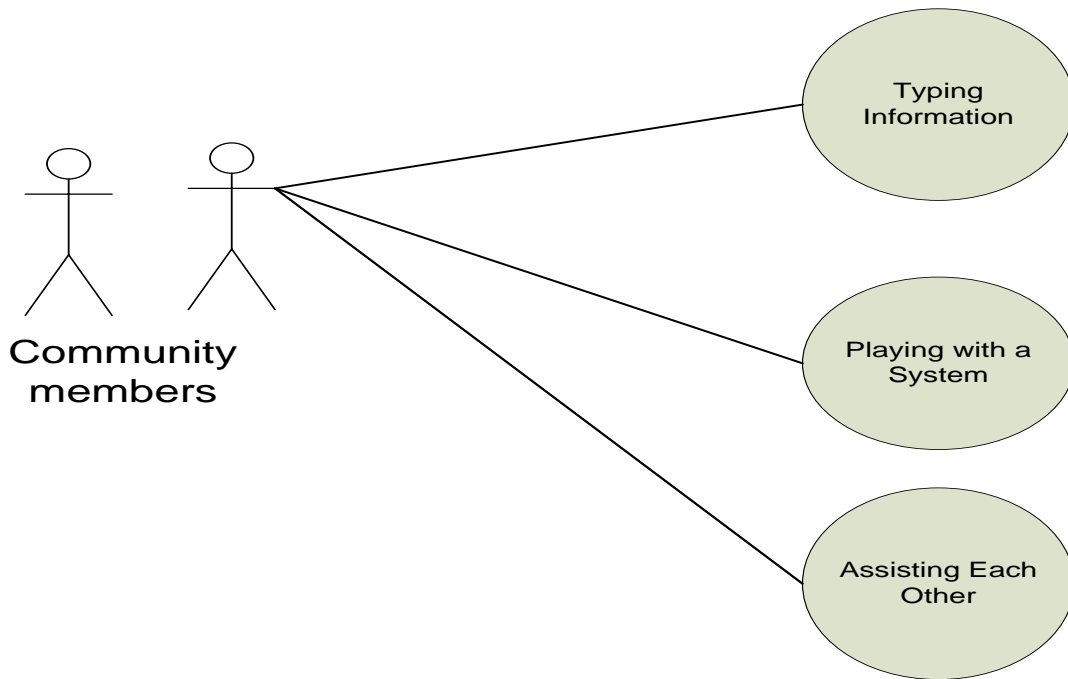


Figure 3-4: Community Members Use Case Diagram

3.3.4 Students

The students and community members have similar authority in the system. These two system users have no administrator rights. The students will use the system for educational purposes. Since the system is responsible for converting text to speech, the students can upload an isiXhosa textbook which the system will read for him/her. Students can learn different commands on Festival speech synthesis so as to advance their technological skills. Learning commands can also be helpful to community members because, in the absence of the administrator, the students can take part in assisting people who are learning how the system works. The other functions that the students perform, since a command line is being used, are that they can change font into different sizes and change the background of the terminal. Figure 3-5, shows the user case diagram of student.

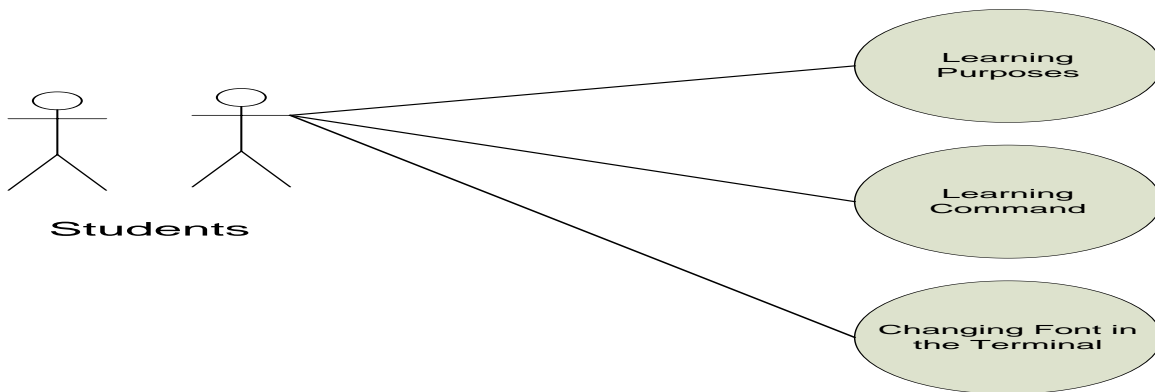


Figure 3-5: Students User Case Diagram

3.4 System Architecture

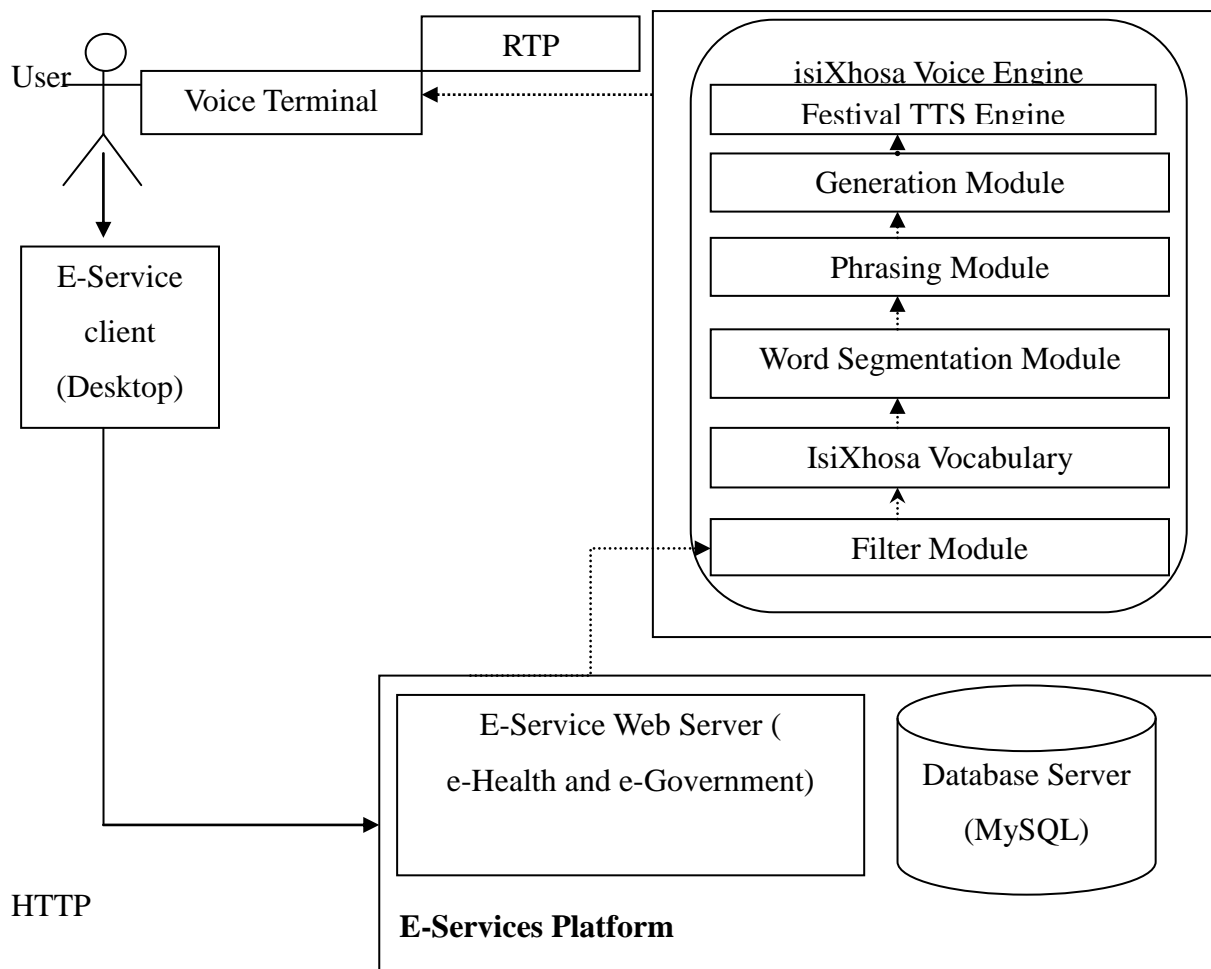


Figure 3-6: System Architecture

Figure 3-6 shows the system architecture of the research and it identifies the connection between the isiXhosa modules and the eService projects. This architecture shows the back-end system design of the system being implemented. According to Figure 3-6, participants or users enter the text as input using a device such as a laptop or desktop. The text can be a single word or a couple of isiXhosa sentences.

There are e-Services projects which already exist in marginalized areas. The e-Service client, e-Service web server and a database server in Figure 3-6 are the part of the network infrastructure which already exists in the Dwesa area. The e-Service client can be a desktop or a laptop or even a mobile phone. In the implementation of this e-Service system the desktop was used as a client. The desktop is used as a device to transmit information into an e-Service web server via Hypertext Transfer Protocol (HTTP). HTTP was used to send a request and transmit files into a web server and MySQL database.

The MySQL database is one of the most popular open-source database systems that are used all over the world. The web server and database server in Figure 3-6 are called e-Service platform. The web server contains all the e-Service projects that are already deployed in the Dwesa area, such as e-Government and e-Health. During the implementation of the Text-to-Speech system for marginalized rural areas, the database called MySQL was created. The e-Service client to e-Service platform was already implemented in previous projects. The isiXhosa Text-to-Speech system was implemented to validate the functionality of these e-Service projects. The Text-to-Speech system will be integrated into the e-Service platform so that it can improve the usability of these services. According to Figure 3-6, there is a link between e-Service and the isiXhosa modules that are being implemented. From the filter module to the voice terminal, each component is designed for a Text-to-Speech system.

3.4.1 Filter Module

The filter module was implemented to filter the data that is sent or received by the server. According to Figure 3-6, the filter module was implemented to filter in and out the sent information from the e-Service platform that contains a web server and a database. In the e-Service client, the user or a participant can type words whether in English or isiXhosa, then

words go straight into a web server or database. The filter module was implemented to allow only isiXhosa words to pass to the isiXhosa modules.

3.4.2 IsiXhosa Vocabulary

The isiXhosa vocabulary module contains isiXhosa words and sentences. The information about the words and sentences was extracted from an isiXhosa dictionary. The information about how to write the voice and silence words was collected from the African Languages and isiXhosa department at the university. IsiXhosa contains some clicks; the position of the tongue is different for each click. The vowels and consonants were written under the phonetic analysis.

3.4.3 Word Segmentation Module

This module was implemented so that it can tokenize a string of text into words. Word segmentation module is dependent on a language in order to offer the best accuracy of a speech. This module is taking words straight from the isiXhosa vocabulary in order to make up similar sentences.

3.4.4 Phrasing Module

The phrasing module was implemented for the isiXhosa Text-To-Speech system to find out the boundaries of phrases. The phrasing module usually checks the beginning and the end of the sentence. The phrasing module is explained in further detail in Chapter 4 of this research.

3.4.5 Generation Module

The generation module was implemented to produce waveforms from the words in a database or from the isiXhosa vocabulary. This module generates waveforms as an output of the text-to-speech system. All of these entire modules are forming the text-to-speech system for the isiXhosa language which was implemented for rural areas. Therefore, after the implementation of these modules, they were uploaded into the Festival engine. After the voice was uploaded, the isiXhosa voice engine was formed. Real-Time Protocol was used to transmit the information speech into a voice terminal. The text comes out as a speech in a voice terminal. This was implemented as a back-end design of the text-to-speech system.

3.5 System Front-end Design

The front-end design usually refers to the website page where information about a website is displayed. But, the text-to-speech system refers to something different from a website but a terminal page that was used to run a system. This terminal page is available on the Linux operating system. The text-to-speech system was implemented to run on top of this application (terminal page).

3.6 Front-end System Architecture

The front-end architecture is the side of the system upon which the participants operate. For example, it is in this system where the participant types words or sentences. Figure 3-7 shows the front-end system design.

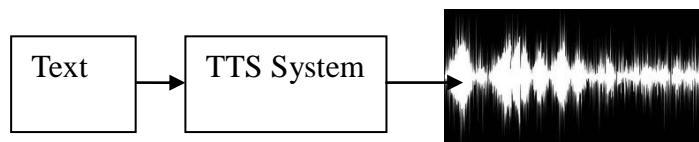


Figure 3-7: General outline of TTS System

Figure 3-7 offers the general outline of a Text-to-Speech system which is being implemented for marginalized rural areas of Dwesa. This represents the front-end system of the Text-to-Speech synthesis. Participants were asked to type in a text using an e-Service client such as a desktop or a laptop. On the system, a raw text was typed with a participant using the Festival synthesis command such as SayText “Molweni“. Therefore, the text is regarded as an input because it is the first step on this system. The text passes through the Text-to-Speech system where the modules are kept on the application. The text changes from being raw text into something that is ready to be rendered as a speech. After the text is processed it comes out as a waveform. The waveform is a last stage of the Text-to-Speech system that was implemented for isiXhosa speakers.

3.7 Design Considerations of the System

During the design of the Text-to-Speech system, there were so many factors that were considered in order for the system to be successfully implemented. These include the naturalness with which the system is able to pronounce the isiXhosa clicks, usability of the system and the system fit in the Dwesa network. The free and open-source software was chosen to model the system. Since the design and development of a database needs an administrator for security reasons. The administrator side was designed and implemented for the maintenance of the system. The recording environment and speaker were considered during a design stage, this is discussed in Chapter 4 of this research. Without forgetting what kind of people we are dealing with in Dwesa, the system uses the cheapest device which is a computer. The problems of people who live in Dwesa were also identified.

3.8 Conclusion

This chapter has offered an overview of how the Text-To-Speech system was designed and its architecture. The components which form the back-end and front-end designs were described here. The database connection and the modules of Text-to-Speech were discussed. The system consideration during the design was explained. The next Chapter will discuss the implementation of Text-To-Speech system.

4 CHAPTER 4: IMPLEMENTATION

4.1 Introduction

In this chapter, the implementation of the Text-To-Speech system for isiXhosa speakers who are living in marginalized rural areas is discussed. In Section 4.2, the system implementation is presented, in Section 4.3, the stages of the Text-to-Speech system implementation are presented. In Section 4.4, the recording of a voice and, in Section 4.5, the Text-to-Speech system database is presented. In Section 4.6, the database tables are presented, in Section 4.7, details of the system administrator are provided. In Section 4.8, the stages of Festvox are presented, and in Section 4.9, the applications of Festival are presented. In Section 4.10, the chapter is concluded with a presentation of the community members who use the Text-To-System system.

4.2 System Implementation

The isiXhosa Text-To-Speech system was implemented using the Festival speech synthesis scheme. The Festival scheme includes modules such as phonest, intonation and phrasing. These modules were connected to the database, so that they can use words and sentences from the database. The modules were implemented for different functionalities of the system. After each stage of implementation the system was shown to users who are teachers, students and community members. The main reason for showing or including users was to avoid disappointment at the end of implementation.

4.3 Text-To-Speech System Stages

The system was implemented in three phases. These include: recording of a voice, database and Festvox.

4.4 Recording of a Voice

We recorded isiXhosa voice phrases and words that we associated with isiXhosa text in this phase of the Text-to-Speech system. The recording was done using a Xhosa male voice during this process. There are some factors that need to be considered before the recording is conducted,

such as choosing a speaker and the environment where the recording will taking place, which are both very important.

4.4.1 Speaker

During the recording, a trained speaker is preferable because s/he is used to the environment but the disadvantage is that these individuals are very expensive compared to general speakers. All these factors were considered before the speaker was chosen. Thus, an untrained speaker was used during the recording process. The speaker was capable of pronouncing all the isiXhosa language vowels and consonants correctly. The speaker was not trained, but his voice is very natural and intelligible. The recording was done during the day when the voice of the speaker was very natural sounding. The environment of the recording place is very important in order to prevent background noise.

4.4.2 Recording Environment

Even if the speaker is audible or is good during the recording, the environment the environment plays a significant part in the success of the recording. There should be no noise in the background, so it should be done in a quiet place such as a recording studio. Since there is no recording studio in Alice, a video conference room in computer science department at the University of Fort Hare was used as the recording room. The microphone was set up properly, so it was easy for the speaker to use it. Therefore, after recording the voice, the information was linked to database tables that contain almost all the necessary isiXhosa language information. The Letter-to-Sound module is connected directly to the MySQL database, as shown in Figure 3-1.

4.5 Database

There are different types of databases such as MySQL, Oracle, Ms SQL, Ingress and many others; however, for this system MySQL was chosen. The reasons why the MySQL database was chosen is that it is free open-source, easy to use, and is good for scalability and performance.

Since it is free open-source software, it is affordable for rural areas. MySQL is defined as a relational database management system that is useful and successful for the long-term management of information and is used in over 4 million installations across the world (Scott, 2010). The information about Text-to-Speech, which is developed for the rural areas of the Eastern Cape, will be captured and stored in a MySQL database called *TTS_db*. This information in MySQL is stored in a database called tables. A table is a collection of related data entries and it consists of columns and rows.

The isiXhosa words and sentences were collected using an isiXhosa dictionary. Two tables of words and sentences were created. The database involves five tables, namely *words*, *sentence*, *month*, *days* and *time* tables. The word table contains the isiXhosa words and the sentence table contains the isiXhosa sentences. The month table contains information about the month of the year, written in isiXhosa. The day table contains information about the days of the week, written using the language which is being used in Dwesa. The time table contains information about time, written in isiXhosa. Listing 4-1 shows the connection between database tables.

```

(cond (ufh_isiXhosa_siphe_diphone_di_16k      (require 'ufh_isiXhosa_siphe_di)
(setup_ufh_isiXhosa_siphe_diphone_lpc_16k_grouped))
(ufh_isiXhosa_siphe_diphone_di_8k          (require 'ufh_isiXhosa_siphe_di)
(setup_ufh_isiXhosa_siphe_diphone_lpc_8k_grouped)) ((and (eq kal_sigpr 'psola)
(eq ufh_isiXhosa_siphe_diphone_groupungroup 'group))
(set!ufh_isiXhosa_siphe_db_words(us_diphone_init ufh_isiXhosa_siphe_psola_group)))
((and (eq kal_sigpr 'psola) (eq ufh_isiXhosa_siphe_diphone_groupungroup
'ungroup)) (set! ufh_isiXhosa_siphe_db_sentence
(isiXhosa_diphone_init ufh_isiXhosa_siphe_psola_sep))) ((and (eq kal_sigpr 'lpc)
(eq ufh_isiXhosa_siphe_diphone_groupungroup 'group)) (set!
ufh_isiXhosa_siphe_db_days (us_diphone_init kal_lpc_group))) ((and (eq kal_sigpr 'lpc)
(eq ufh_isiXhosa_siphe_diphone_groupungroup 'ungroup)) (Set!
ufh_isiXhosa_siphe_db_month

```

Listing 4 1: Database connection to modules

Listing 4.1 shows the connection between database tables and the Text-to-Speech Modules. These tables are also connected into a LPC group that is the part of the module.

4.6 Database Table

The information from the Dwesa Text-to-Speech system will be captured and stored in the MySQL database called *TTS_db*. The information will be stored in *TTS_db* in the form of tables. All the tables contained in the *TTS_db* database will be presented in the ensuing sections of this study.

4.6.1 Words Table

The words database table stores information about the isiXhosa words which form part of the database. The words table contains different types of words; some have clicks, while some are voiceless and unvoiced words. The words were collected in such a way that when the user requests a word from the database, the word must be available. This table contains more than 5000 words. Table 4-1 shows the words table with its field and descriptions.

id	words
1	mhlana
2	qaqamba
3	molweni
4	kunjani
5	bhuti

Table 4-1: Words Table

4.6.2 Sentence Table

The sentence database table captures and stores information about isiXhosa sentences which are used to communicate with isiXhosa speakers. The sentences were collected using an isiXhosa dictionary. Most of the sentences which are available in the database tables are those which are commonly used in marginalized rural areas, such as *Molweni*, *unjani*. This database table contains an id field and a sentence field. Table 4-2 shows the database table which contains isiXhosa sentences.

id	sentence
1	molweni ekhaya, ninjani namhanje
2	Igqirha lendlela nguqongqothwane
3	Ebeqabele egqithapha unguqongqothwane

Table 4-2: Sentences Table

4.6.3 Month Table

The month database table stores information about the month of the year, written in isiXhosa. This information will help illiterate people to know the month of the year. This table consists of an id field and a month field. Table 4-3 shows the table which contains all the months of the year from January to December.

id	month
1	EyoMqungu
2	EyoMdumba
3	EyoKwindla

Table 4-3: Month Table

4.6.4 Days of the Week Table

This table stores information about the days of the week, written in the language which is used in marginalized rural areas, in this case isiXhosa. This information will remind the isiXhosa speakers about how to pronounce the days of the week in isiXhosa. Table 4-4 shows the table that contains the days of the week.

id	days
1	uMvulo
2	uLwesibini
3	uLwesithathu

Table 4-4: Days of the Week

4.6.5 Time Table

This database table stores information related to time. There is a command on Festival speech synthesis that can be used so that the system will say the time. This information will help the user who, at the present time, has a watch or a cell phone to check the time. However, in this case, the information will be written using the isiXhosa language. This information will be helpful to illiterate people because if someone is illiterate he/she cannot use a watch to check the time. Table 4-5 shows the table with time information.

id	time
1	ixesha yintsimbi yesithathu
2	ixesha yintsimbi yesibini
3	ixesha yintsimbi yesine

Table 4-5: Time Table

These database tables contain different types of information whilst trying to avoid a lot of information being packed together in one table. All these tables are linked together with a common field called an id.

4.7 System Administrator

As has already been presented in Chapter 3, the administrator is the overall manager of the Text-to-Speech system. The administrator is responsible for any action that is taken in the system by users. The administrator has all the authority of the system, for instance; in this case the administrator runs the code that controls the entire Text-to-Speech system module. The entire module responds if there is any error. Furthermore, the administrator can add and/or remove information from the database. He/she can also assist users in how to use the system. Listing 4-2 shows the administrator code.

```
(Defvar ufh_isiXhosa_siphe_diphone_dir (cdr (assoc 'ufh_isiXhosa_siphe_diphone voice-
locations)) "ufh_isiXhosa_siphe_diphone_dir The default directory for the ufh isiXhosa siphe
diphone database.")(set! load-path (cons (path-append ufh_isiXhosa_siphe_diphone_dir
"festvox/") load-path))(require 'radio_phones)
  (require_module 'UniSyn);; set this to lpc or psola(defvar
ufh_isiXhosa_siphe_diphone_sigpr 'lpc);; Rset this to ungroup for ungrouped
version(defvar ufh_isiXhosa_siphe_diphone_groupungroup 'group)(if (probe_file (path-
append ufh_isiXhosa_siphe_diphone_dir "group/kallpc16k.group")) (defvar
ufh_isiXhosa_siphe_diphone_index_file (path-append
ufh_isiXhosa_siphe_diphone_dir "group/kallpc16k.group")) (defvar
ufh_isiXhosa_siphe_diphone_index_file (path-append
ufh_isiXhosa_siphe_diphone_dir "group/kallpc8k.group")) (set!
ufh_isiXhosa_siphe_diphone_psola_sep (list '(name
"ufh_isiXhosa_siphe_diphone_psola_sep") (list 'index_file (path-append
ufh_isiXhosa_siphe_diphone_dir "dic/diphdic.est")) '(grouped "false") (list
'coef_dir (path-append ufh_isiXhosa_siphe_diphone_dir "pm")) (list 'sig_dir (path-
append ufh_isiXhosa_siphe_diphone_dir "wav")) '(coef_ext ".pm") '(sig_ext
".wav"))
```

Listing 4-2: System Administrator

Listing 4-2 shows how the system administrator communicates and interacts with the database tables and Text-to-Speech system modules.

4.8 Festvox

Festvox contains the suite of tools that are used to build a synthetic voice for Festival speech synthesis. In other words, Festvox works under Festival in order to produce a natural sounding voice. The implementation of Festvox includes more modules of speech synthesis such as phoneset, duration, etc. These modules were implemented so that a natural sounding voice should become an output at the end of the development of the system. In order for those modules to work properly they need a piece of code that can link them all. The code that puts all these modules together was implemented. The listing that was created to link other modules is shown in Listing 4.3:

```
(defvar ufh_isiXhosa_siphe_diphone_dir
  (cdr (assoc 'ufh_isiXhosa_siphe_diphone voice-locations))
  "ufh_isiXhosa_siphe_diphone_dir
  The default directory for the ufh isiXhosa siphe diphone database.")
(set! load-path (cons (path-append ufh_isiXhosa_siphe_diphone_dir
  "festvox/") load-path))
(require 'radio_phones)
(require_module 'UniSyn)
;; set this to lpc or psola
(defvar ufh_isiXhosa_siphe_diphone_sigpr 'lpc)
```

Listing 4-3: Listing that link other modules

Listing 4.3 is written with a scheme which is used to link all the modules that were implemented. This listing was implemented in such a way that it works as an engine to control other schemes. In order to find whether there is any error during the implementation of other schemes we just compile this listing module. The command line (Terminal) is used to compile the code. One will find that all the listings in the scheme end up with .scm. Therefore, this listing goes straight to the first module that needs to be implemented. The phoneset scheme of the isiXhosa language was implemented.

4.8.1 Phoneset Module

Phoneset is a set of symbols which may be further defined in terms of their features, such as vowels and consonant. Phoneset is the most important module that was being implemented because the entire other module depends on the phoneset. If the system understands this module, it is easy for other modules of speech synthesis. The phoneset module is very important when you develop a new language in the Text-To-Speech synthesis. This module was implemented in such a way that it can connect with other modules in order to produce a natural sounding voice. Every module has its definition during its implementation. Phoneset was defined as follows:

```
(defPhoneset
  NAME
  FEATUREDEFS
  PHONEDEFS
)
```

The defPhoneset is the definition of the phoneset module; this definition was given a unique symbol to differentiate it from other modules. FeatureDefs is a list of definitions which each consist of a feature name and its possible values. The vowels and consonants are defined in the implementation of a phoneset module. Since the phoneset is the first module that needs to be implemented during the development of a new language, the vowels and consonants were implemented first so that the system can understand them and know how to pronounce them. For example:

```
(
  (VC +-
    (VLeng short long diphthong schwa O));; vowel
  .....
)
```

The V and C stand for isiXhosa vowels and consonants. The vowels and consonants were implemented in such a way that they have what we call silence phones. The silence phones identify the end of the sentence. This was done during the implementation, through the command *PhoneSet.silence*. The # sign signifies the beginning of the sentence, or sometimes the silence of the sentence. The phoneset module contains different types of vowels and consonants that were

being implemented in this system, such as nasal sounds and clicks. The Listing 4.4 was created to show how the phoneset for isiXhosa was implemented:

```
(defPhoneSet
ufh_isiXhosa_siphe_diphone
;;; Phone Features
(
;; vowel, consonant, diacritic, silence, closure or other
(vc + - d s cl 0)
;; vowel length: shrt long diphtong schwa
(vlng s l d a 0)
;; vowel height: high mid low
(vheight 1 2 3 0)
;; vowel frontness: front mid back
(vfront 1 2 3 0)
;; lip rounding
(vrnd + - 0)
;; consonant types : [p] stop (plosives), [f] fricative, [h] affricate,
;;                    [a(l/r/g/o)] approximant(lateral/retroflex/gluide/other), [n]
nasal,
;;                    [c] clicks, [t] tap/flap, [r] trill, [v] voiced implosives,
;;                    [sa] stop with aspiration, [ca] click with aspiration, [la]
"lateral affricate",
;;                    [nc] nasalized clicks
(ctype s f a al ar ag ao n c t r v sa ca la nc 0)

(PhoneSet.silences '(pau # H#))
(define (ufh_isiXhosa_siphe_diphone::select_phoneset)
  "(ufh_isiXhosa_siphe_diphone::select_phoneset)
  Set up phone set for ufh_isiXhosa_siphe_diphone."
  (Parameter.set 'PhoneSet 'ufh_isiXhosa_siphe_diphone)
  (PhoneSet.select 'ufh_isiXhosa_siphe_diphone)
)
(define (ufh_isiXhosa_siphe_diphone::reset_phoneset)
  "(ufh_isiXhosa_siphe_diphone::reset_phoneset)
  Reset phone set for ufh_isiXhosa_siphe_diphone."
  t
)
)
(provide 'ufh_isiXhosa_siphe_diphone_phoneset)
```

Listing 4-4: Phoneset listing for the isiXhosa language

The Listing 4.4 starts with the definition of the phoneset module; this is followed by the name of the language that was implemented. After that, all the features of the language were implemented

- such as vowels and consonants - and the different types of phoneset were also identified in this piece of code. This is the structure of how the phoneset was implemented for isiXhosa. The phoneset module of the isiXhosa system is defined as: *ufh_isiXhosa_siphe_diphone_phoneset.scm*. Since the phoneset module is connected to a database that contains almost all the isiXhosa language words, there is another module that provides every word which is not in the database. This module is called the Letter-To-Sound rule (LTS). Therefore, if the word is not in a database, Festival is not able to pronounce it; instead it just spells the word. The letter-to-sound rule was implemented to assist the phoneset and the database.

4.8.2 Letter-To-Sound Module

The letter-to-sound module is only responsible for the determination of the phonetic transcription of the coming text. It was difficult to implement this module because the isiXhosa language has some words that correspond to several entries in the dictionary, but with different pronunciations. The letter-to-sound rule was implemented as a backup so that when the word is not in the database, it just constructs the word from the existing vowels and consonants from the database. Since the system fails to pronounce it if it is not in the database, the LTS is there to generate that word. Listing 4.5 was implemented to show how letters were mapped to form a word in the letter-to-sound rule for the isiXhosa language.

```

(Its.ruleset
  ufh_isiXhosa_siphe_diphone
    ( (Vowel a e i o u) )
    (
      ;; LTS rules
      ([a]=a)
      ([e]=e)
      ([i]=i)
      ([o]=o)
      ([u]=u)
      ( [ "" a ] = a )
      ( [ "" e ] = e )
      ( [ "" i ] = i )
      ( [ "" o ] = o )
      ( [ "" u ] = u )
      ( [ "" m ] = mm )
      ( [ "" n ] = nn )
      ( [ "-" a ] = a )
      ( [ "-" e ] = e )
    )
  )

```

Listing 4-5: Letter-To-Sound mapping rules

Listing 4.5 shows how the vowels and consonants were mapped in letter-to-sound to come up with constructed words or sentences. Since the scheme listings on Festival are too long, only the section that shows the mapping of words is offered here. The letter-to-sound rule is named on the system as *ufh_isiXhosa_siphe_diphone_lts_rules.scm*. Letter-to-sound rules lead to the most important module which is called the phrasing module.

4.8.3 Phrasing Module

The database was well labeled; in order for a phrasing module to work properly the database needed to be well labeled. This module involved different types of trees that were implemented to denote the end of the utterance. Listing 4.6 shows how the CART tree was implemented to predict a simple break due to punctuation:

```

(set! isiXhosa_phrase_CART_tree
  „((lisp_token_end_punc in (“?” “.” “:.”))
    ((BB))
      ((lisp_token_end_punc in (““” “\” “,” “.”))
        ((B))
          ((n.name is 0)); end of utterance
            ((BB))
              ((NB))))))

```

Listing 4-6: CART Tree Module

Listing 4.6 starts by defining the name of the tree, in this case: set! isiXhosa_phrase_CART_tree. The alphabet in this piece of code represents different symbols, for example, B stands for a short break in the sentence, BB stands for long break and NB stands for no break at all. These symbols were put into the system implementation because, during punctuation, these breaks are very important. Listing 4.7 shows how the phrase module was implemented for the isiXhosa language.

```

((lisp_token_end_punc in ("?" "." "!" ))
  ((BB))
    ((lisp_token_end_punc in (";" ":" "-" "--"))
      ((B))
        ((lisp_token_end_punc in (","))
          ((B))
            ((n.name is 0) ;; end of utterance
              ((BB))
                ((NB))))))
  (define (ufh_isiXhosa_siphe_diphone::select_phrasing)
    "(ufh_isiXhosa_siphe_diphone::select_phrasing)
    Set up the phrasing module for isiXhosa language."
    (set! phrase_cart_tree ufh_isiXhosa_siphe_diphone_phrase_cart_tree)
    (Parameter.set 'Phrase_Method 'cart_tree)
    (Param.set 'Phrasify_Method Classic_Phrasify)
  )
  ( define (ufh_isiXhosa_siphe_diphone::reset_phrasing)
    "(ufh_isiXhosa_siphe_diphone::reset_phrasing)
  t

```

Listing 4-7: Phrasing Module with types of Breaks

Listing 4.7, above, links to the punctuation program and CART tree code because these modules work hand-in-hand. Without a proper labeled database and proper punctuation it is impossible for a phrasing module to work. The phrasing module was given a name: `ufh_ishXhosa_siphe_diphone_phrasing.scm`. Therefore, after the punctuation and phrasing were implemented successfully, these lead us to another very important module called Intonation Analysis.

4.8.4 Intonation Analysis

This module was implemented solely to determine a tone or pitch contour from the vowels and consonants, or from isiXhosa sentences. The end result of determining the tone was that, in the isiXhosa language, tone rises in a question and drops down at the end of a statement. Listing 4.8 presents the way in which the tone is determined from vowels and consonants:

```
(set! ufh_isiXhosa_siphe_diphone_accent_cart_tree
'
(
(R:SylStructure.parent.gpos is content)
( (stress is 1)
((Accented))
((NONE))
)))
(define (ufh_isiXhosa_siphe_diphone::select_intonation)
"(ufh_isiXhosa_siphe_diphone::select_intonation)
Set up intonation for isiXhosa."
(set! int_accent_cart_tree ufh_isiXhosa_siphe_diphone_accent_cart_tree)
(Parameter.set 'Int_Target_Method 'Simple)
)
(define (ufh_isiXhosa_siphe_diphone::reset_intonation)
)
(provide 'ufh_isiXhosa_siphe_diphone_intonation)
```

Listing 4-8: Determination of the tone from vowels and consonants

Listing 4.8; shows that the intonation module identifies the tone or the accent on vowels and consonants using the CART tree. This module was also implemented in such a way that it can predict the stress from sentences or vowels. In listing 4.8 we find that the stress was set to

number 1. It only targets to predict those vowels and consonants or sentences that containing high levels of stress; that is why we have the number 1 in this code, and not zero. This intonation module was named `ufh_isiXhosa_siphe_diphone_intonation.scm` on the system. Every listing on Festival ends with *.scm* meaning that it was developed using the Festival schemes. After the intonation module was developed to predict the tone of the sentence, the researcher was led to the prediction of the length or speed of the speech.

4.8.5 Duration Parameter Module

Since Festival speech synthesis has different types of tree methods, the prediction of speed from vowels and consonants, the zscores tree, was used to make this module successful. Listing 4.9 shows how the zscores were used in this system in order to predict speed.

```
(set! ufh_isiXhosa_siphe_diphone::zdur_tree
  '
  ((R:SylStructure.parent.R:Syllable.p.syl_break > 1 )
    ((1.5))
  ((R:SylStructure.parent.syl_break > 1)
    ((1.5))
  ((1.0))))
(set! ufh_isiXhosa_siphe_diphone::phone_durs
  '(
  ;;; PHONE DATA
  ;; name zero mean in seconds e.g.
  (# 0.0 0.250)
  (gq 0.0 0.25)
  (q 0.0 0.25)
  (nc 0.0 0.25)
  (c 0.0 0.25)
  .....
```

Listing 4-9: Zscores tree prediction of duration

This kind of tree signs each phone a duration number, thus, we find that all the clicks in listing 4.9 have the same numbers. Listing 4.9 for zscores was named in the system as *ufh_isiXhosa_siphe_diphone_durdata.scm*. The zscores tree code was linked direct to the duration model code that was used to make the system speak very slowly. Listing 4.10 shows

the commands that were used to control the system.

```
(require 'ufh_isiXhosa_siphe_diphone_durdata)

(define (ufh_isiXhosa_siphe_diphone::select_duration)
  "(ufh_isiXhosa_siphe_diphone::select_duration)
  Set up duration model."
  (set! duration_cart_tree ufh_isiXhosa_siphe_diphone::zdur_tree)
  (set! duration_ph_info ufh_isiXhosa_siphe_diphone::phone_durs)
  (Parameter.set 'Duration_Method 'Tree_ZScores)
  (Parameter.set 'Duration_Stretch 1.1)
  )
(define (ufh_isiXhosa_siphe_diphone::reset_duration)
```

Listing 4-10: Duration module that linked to the zscores tree

Listing 4.10, that is the duration module linked to the zscores, is controlled with the commands in order for the system to speak fast or slowly. The command is `Parameter.set 'Duration_Stretch`. This command depends on the changing of numbers, like in listing 4.10 the number is 1.1 which means the system speaks fast. Therefore, if you increase the number to 5.5 the system pronounces the words very slowly. The duration module for isiXhosa text-to-speech was named ***ufh_isiXhosa_siphe_diphone_duration.scn***. This module leads us to one of the very important modules that was implemented so as to predict where the accent goes.

4.8.6 Fundamental Frequency (F0) Generation

This type of module was implemented for the isiXhosa language in order to predict the pitch and accent of the language. F0 module was implemented in order to improve the quality of the speech synthetic of the isiXhosa language. F0 module is divided into two approaches, the first one is the statistical approach and the second is the rule based approach. Therefore, the rule based approach was used to implement this system. Listing 4.11 shows how the F0 module was implemented.

```

(define (ufh_isiXhosa_siphe_diphone::select_f0model)
  "(ufh_isiXhosa_siphe_diphone::select_f0model)
  Set up the F0 model for ufh_isiXhosa."
  (Parameter.set 'Int_Target_Method 'Int_Targets_Default)
  )
(define (ufh_isiXhosa_siphe_diphone::reset_f0model)
  "(ufh_isiXhosa_siphe_diphone::reset_f0model)
  Reset F0 model information."
  t
  )
(provide 'ufh_isiXhosa_siphe_diphone_f0model)

```

Listing 4-11 : Fundamental Frequency Module code

Listing 4.11 shows how the rule based approach was used in Festival to implement the new Text-to-Speech system for the isiXhosa language. The final stage of the implementation was waveform generation. Listing 4.11, shows more how the waveform was created as an output of the system.

4.8.7 Waveform Generation Module

The waveform module uses a pre-recorded voice of the language that is being developed. The Text-To-Speech system for isiXhosa was implemented using a concatenative approach which results in the very natural sounding voice of the system. It was mentioned in Chapter 1 that the first step of Text-To-Speech is inputting text into a system. Therefore, waveform generation is the final stage of Text-To-Speech architecture in that it produces the output as a speech.

As previously discussed, in section 4.3.2 about the database system that was being developed which contains all the isiXhosa words, when there is a database system an administrator is required to be there to control the system. The administrator should be responsible for editing, modifying and assisting participants during registration on the database. There is no registration of participants neither are they required to put their confidential information onto the system. Therefore, there is no need for an administrator because it was implemented for rural community members and anyone who knows how to use a computer can edit the database. The participants communicate with the system by typing the isiXhosa words on the Festival terminal (command

line). There is no need to have a surname and password in order to have the authority to use the system.

4.9 Festival Application

This discussion of the text-to-speech system dwells on how Festival speech synthesis works. Figure 4-1 shows how the path indicating where the voice is allocated on Festival is found.

```
mhlana@siphe:~$ cd /usr/share/festival/voices/xhosa/ufh_isiXhosa_siphe_diphone/festvox/  
mhlana@siphe:/usr/share/festival/voices/xhosa/ufh_isiXhosa_siphe_diphone/festvox$
```

Figure 4-1: Voice path

Therefore, if the path is correct, the open-source software called Festival synthesis starts to run the application. You find that if the music is being played while you are trying to compile the application on terminal command, the error message pops up to say **Linux: can't open /dev/dsp**. This means that two applications which are in need of a voice cannot be run at the same time. Figure 4-2 shows the type of error displayed.

```
diphone/festvox$ festival  
Festival Speech Synthesis System 1.96:beta July 2004  
Copyright (C) University of Edinburgh, 1996-2004. All rights reserved.  
For details type `(festival_warranty)'  
festival> (Say  
SayPhones SayText  
festival> (SayText "Molweni")  
Linux: can't open /dev/dsp  
#<Utterance 0xb6a52db8>  
festival>
```

Figure 4-2: The error when you are playing music on the background

Festival speech synthesis runs this system with the following command: **festival** > after which the text that the participants want to hear is written as speech, by expending this command: **festival** > **(SayText "Molweni Molweni bhuti")** (Hello Hello my brother). Therefore, after this line of command the participants, or users, hear the output as speech. Figure 4-3, shows the

command that is being used.

```
diphone/festvox$ festival
Festival Speech Synthesis System 1.96:beta July 2004
Copyright (C) University of Edinburgh, 1996-2004. All rights reserved.
For details type `(festival_warranty)'
festival> (Say
SayPhones SayText
festival> (SayText "Molweni")
Linux: can't open /dev/dsp
#<Utterance 0xb6a52db8>
festival> (SayText "Molweni Molweni Bhuti")
#<Utterance 0xb6c93ab8>
festival>
```

Figure 4-3: Festival Commands

There are different commands on Festival speech synthesis such as SayText and SayPhones that present all the language phones.

4.10 Community Members using the system

The Text-To-Speech system was implemented for Dwesa community members so as to improve their skills in using a computer. It was previously stated that there is no need for the participants (community members) to have authorization in order to access the system. The only thing that was required from the participants was that they must have a computer skill or must have used a computer before. The system was implemented for isiXhosa speakers who live in the Dwesa rural community, where most people cannot write and read in their mother tongue. The developer was forced to localize the system.

4.11 Conclusion

The chapter started by introducing and discussing the implementation of the Text-To-Speech system for the rural area of Dwesa. It continues by discussing the stages of the Text-to-Speech system and its recording process. The database and database tables were presented in this chapter. The authority of the administrator on the system and the festvox stages were presented in this chapter. Festival application, and how it works with speech, was explained. The chapter concluded by presenting the system used by rural community members. In the next chapter, the testing and results are presented.

5 CHAPTER 5: SYSTEM TESTING AND RESULTS

5.1 Introduction

The system needs to be tested before deployment; a test run of the system was done to remove all the unnecessary information. The testing stage is very important because it shows that the system was successfully developed. The system was demonstrated to three groups of people: community members, students and teachers, since these are the users of the system. The main purpose of involving users in the testing process was to get feedback on the usability of the system. The results of the tests were important in order to match them to the expected results mentioned in Chapter 1. This chapter focuses on the system testing, the results thereof and the evaluation of the results obtained. The main parameters considered when testing the system were the functionality and usability of the system.

5.2 Testing Apparatus

After the system was implemented a test was necessary before deployment to the Dwesa area. During the testing process, different devices were used to make sure that the test was successful. A desktop computer with a good sound card and earphones was used and set properly so that the user can hear what the system was saying. Pen and paper were also the apparatus of the test; the main reason for using this apparatus was that users were given a task to complete. This task entailed that if a user hears a word from the system, the user must write it down as feedback.

5.3 Testing Environment

The test was conducted in the lab and twenty users were tested; users include teachers, students and community members. The volume of the system was well adjusted so that each user could hear what the system was saying clearly. The environment in the lab during the test was calm and there was no background noise. Other computers were logged off to avoid background sounds. The system pronounced the words clearly and the users were able to hear every word.

5.4 Functionality Testing

Functionality testing is defined as a method of assessing the fulfillment of the system's objectives based on

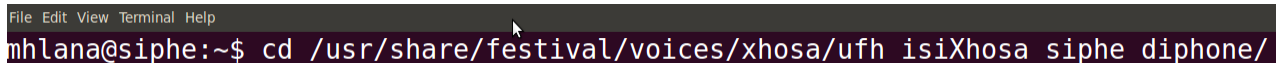
set test parameters. The process may incorporate other methods such as unit testing in which individual system components are subjected to tests in order to verify whether they perform to the set goals.

5.4.1 Components testing

The TTS is composed of components that interact with each other to perform various tasks. These components include:

- ❖ **Phonset** : a set of symbols which may further be defined in terms of features such vowels consonants.
- ❖ **Phrasing** : used to ascertain the boundaries of phrases which are realized by many factors such as pause.
- ❖ **Intonation**: used in the prediction of accent.
- ❖ **Duration**: Used to fix the sizes of all phones.
- ❖ **F0 Generation**: used to predict the accent of the current text supplied at the prompt.
- ❖ **LTS (Letter-to-Sound)**: used to assign pronunciation to words not found in the lexicon, a festival database.

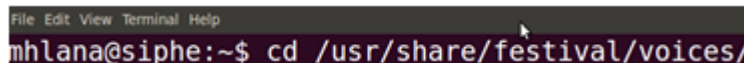
A system run-through was done to demonstrate the system functionality of the above components. One of the notable results of a properly working system was to have a set of chosen isiXhosa words converted to voice. This was demonstrated by the following Figures.



```
File Edit View Terminal Help
mhlana@siphe:~$ cd /usr/share/festival/voices/xhosa/ufh_isiXhosa_siphe_diphone/
```

Figure 5-1: Path where festival voices are stored

Figure 5-1, shows the path where the isiXhosa voice was placed. Other voices are under the *voices* folder shown in Figure 5-2:-



```
File Edit View Terminal Help
mhlana@siphe:~$ cd /usr/share/festival/voices/
```

Figure 5-2: Location of all festival voices

When the system is properly configured, the simplest test to perform is to supply the command *festival* at the shell prompt as shown in Figure 5-3:-


```
mhlana@siphe:/usr/share/festival/voices/xhosa/ufh_isiXhosa_siphe_diphone$ festival
```

Figure 5-3: Festival testing

The expected results of Figure 5-3 are shown in Figure 5-4. Figure 5-4 shows that festival was properly installed and awaits the user to supply the next command which is the actual text to be converted. However, it is also possible to view voices that come with festival including the custom voices, in this case the isiXhosa voice.

```
mhlana@siphe:/usr/share/festival/voices/xhosa/ufh_isiXhosa_siphe_diphone$ festival
Festival Speech Synthesis System 1.96:beta July 2004
Copyright (C) University of Edinburgh, 1996-2004. All rights reserved.
For details type `(festival_warranty)'
festival>
```

Figure 5-4: Festival output

Figure 5-5, shows all the possible voices that are installed on Festival speech synthesis.

```
festival> (voice
There are 152 possibilities. Do you really
want to see them all (y/n) ?
```

Figure 5-5: Displaying voice options

When a user has chosen the desired voice, text is entered in the manner displayed in Figure 5-6. Figure 5-6 shows that the text “Molweni Bhuti” was entered at the prompt and festival through its *phoneset* component which converted the text to voice which was captured by the code #<Utterance 0xb6b24a38> in Figure 5-6.

```
festival> (SayText "Molweni bhuti")
#<Utterance 0xb6b24a38>
```

Figure 5-6: Festival text conversion

One interesting feature of this program is that a user can vary the speed of utterances. This helps to

accommodate all the people with various problems, including those with hearing handicaps. Figure 5-7 shows how to configure the variation on speed of utterance in a given set of words. Parameter 4 indicates the time, in seconds, that a certain word has to be uttered.

```
festival> (Parameter.set 'Duration_Stretch 4)
4
festival> (SayText "Molweni Bhuti")
#<Utterance 0xb6d6a8d8>
```

Figure 5-7: Festival changing of voice speed

5.4.2 Usability Testing

This type of testing was conducted to verify whether the system is appreciated by the intended recipients or users. Since the system was designed for use in the Dwesa community, it was necessary to assess how network users in that area embrace the system. We appreciate that the system does not have a friendly interface in comparison to the normal web applications which are witnessed every day. We also acknowledge the difficulties that come with the use of command driven interfaces. However, we take this as an added advantage to the users as they would be able to gain more hands-on training and ultimately learn various computer skills thereafter.

5.4.2.1 System users

System users included teachers and students from the five schools covered by the project, as well as community members. A total of twenty users were chosen, as shown in Table 5-1.

Table 2: Sample users

User	Gender	Age	Occupation Status	Computer Literacy Level
1	Male	15-20	Unemployed	Average
2	Male	21-30	Student	Very Good
3	Male	21-30	Student	Good
4	Male	21-30	Employed	Poor
5	Male	21-30	Employed	Poor
6	Male	31-40	Unemployed	Average
7	Male	41-60	Pensioner	Poor
8	Female	21-30	Student	Average
9	Female	21-30	Student	Very Good
10	Female	21-30	Student	Very Good
11	Female	21-30	Employed	Good
12	Female	31-40	Employed	Good
13	Female	31-40	Employed	Good
14	Female	41-60	Employed	Poor
15	Female	41-60	Pensioner	Poor
16	Male	15-20	Student	Good
17	Male	15-20	Student	Good
18	Male	31-40	Employed	Good
19	Female	31-40	Employed	Good
20	Male	21-30	Student	Good

5.4.2.2 Training needs

The twenty users shown in Table 5-1 were trained on how to use the system. However, the training had some setbacks as 25% of the users were computer illiterate. As a result, thorough training was conducted first, in order to get these users to the same skill level as the others. As demonstrated in Figure 3-7, users were shown how the system works and they were told what is expected of them from the exercises conducted. At the end of the training exercise, testers were asked to complete a questionnaire. The questionnaire was meant to capture the users' responses on the type of training and their level of understanding of the system components. The results were depicted in Figure 5-8. From Figure 5-8, 80% of the respondents indicated that it was necessary to provide users with basic training skills prior to the use of the system and they seemed to be satisfied with the type of training material used to conduct training. This was captured by the 90% response on the training material parameter. The majority of the users were also happy with the duration of training, which was 30 minutes for each group of ten users. However, a few individuals indicated that they needed more time as it was their first time using a computer. This was mainly the old aged or pensioners.

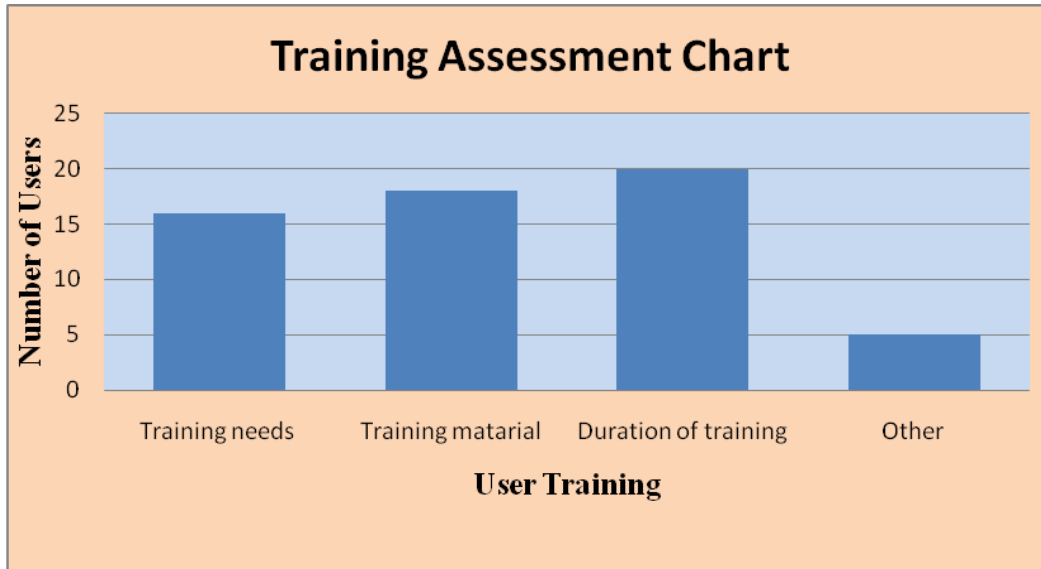


Figure 5-8: Training needs assessment chart

5.4.2.3 Text-To-Speech Conversion

When training was done with the twenty selected users, they were then given sample tasks to select a few sample isiXhosa words for use in the system. Some of the words are shown in Table 5-2.

Table 3: Sample isiXhosa words used by users

ID	IsiXhosa Words
1	Molweni
2	Ninjani
3	ekhaya
4	Qaqamba
5	bhuti
6	Tata

The words in Table 5-2 are classified into two columns as they are represented in the festival

database. The ID is a unique key which identifies each individual word used in the database. When testing, users may not be aware of how these IDs are used but, for the sake of clarity, we used them to demonstrate a hidden concept. The first word “*Molweni*” is an isiXhosa word with an equivalence to the word *hello* in English. The assumption is that when this word is supplied to the program the output would be a sound conversion of the word “*Molweni*”. First users used single words as, shown in Table 5-2, but they later used sentences to generate voices from the text supplied. Listing 5-1 shows one of the paragraphs supplied to the program, which generated interesting results amongst the testers.

“Qongqothwane

Igirha lendlela nguqongqothwane

Igirha lendlela kuthwa nguqongqothwane”

(The witchdoctor of the road is the beetle)

(The witchdoctor of the road is said to be the beetle)

“Ebeqabel egqithapha uquongqothwane

Ebeqabel egqithapha uquongqothwane”

(He has passed by up the steep hill, the beetle)

Listing 5-1: Sample isiXhosa sentences

Although the program did not produce the original lyrics of this song it was interesting to hear the sound of “q” and “ngq” from the sentences supplied to the program. During the test, a hand phone that was plugged into a computer was used. The users were placed in the opposite direction to the computer screen so that they could not see what the developer was typing on the screen. The main reason for doing this was because the users were given the task of writing down every word they heard from the system, without seeing what was written in the command interface. If the user hears nothing from the system, they were given a second chance when the words were repeated. Words like *Molweni* (hello), *ninjani* (how are you) and *bhuti* (brother), were understood in the first attempt because these words are used every day. However, there was a difference when users were listening to words such as *qaqamba* (person’s name) and *qongqothwane* (beetle). Some of the users even took two chances in order to hear what the system. The system can pronounce the isiXhosa clicks and vowels very well.

5.4.2.4 Results of Text-To-Speech conversion exercise

As stated earlier, that test subjects were used to measure the impact of the system on themselves and other users in general; a set of questionnaires was issued to these test subjects. The questionnaire was meant to capture responses pertaining to:

- ❖ The quality of voice produced by the system,
- ❖ The clarity of vowels and consonants from various text supplied to the program,
- ❖ System interface,
- ❖ The relevance of the system to their needs.

The results of the analysis of the responses to the parameters above are given in Table 5.3. Users were asked to rate the parameters on a scale from 1 to 10, where 10 means clear and sufficient while 1 means unclear and needs attention. The figures in Table 5.3 show the average rating of each parameter by all twenty users.

Table 4: System rating results

Parameter	Average rating
Quality of voice	6.3
Clarity of vowels and consonants	7.7
System interface	3.6
Relevance of the system	8.9

Table 5-3 shows the weighted averages of each parameter rounded off to one decimal place. These values were then represented in the form of a graph shown in Figure 5-9.

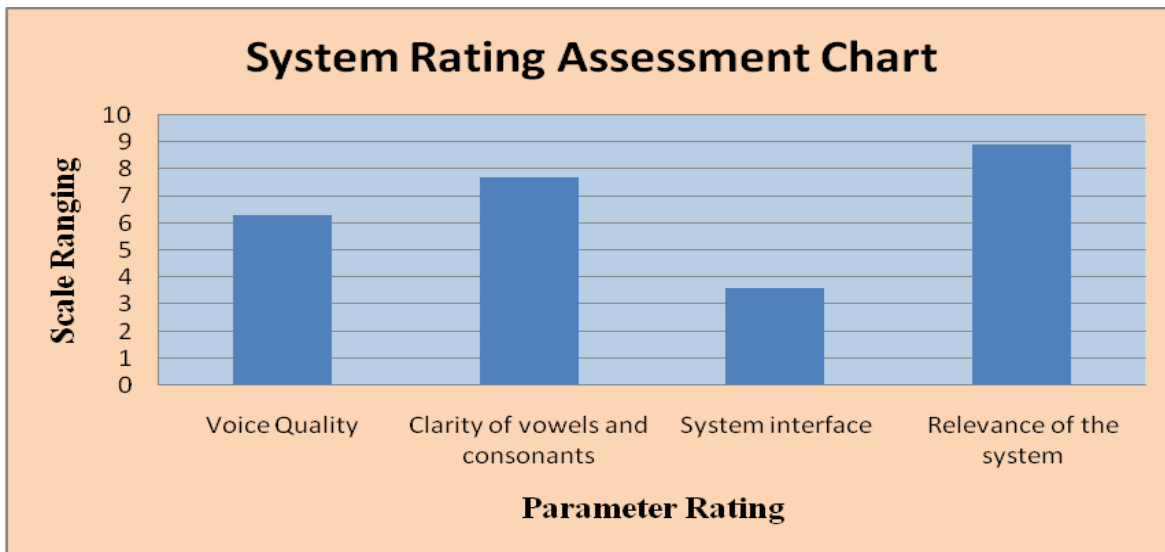


Figure 5-9: System Rating Assessment Chart

According to the analysis of the results of users' responses to the questionnaire, 88% of the respondents appreciated the relevance of the system to their needs. This was evident even during training as most of them kept trying various isiXhosa words and sentences in order to generate the associated sounds. It was also interesting to note that the majority of the test subjects acknowledged the fact that the program produced nearly the exact sounds of the vowels and consonants of the text used. However, they complained about the low quality of voice due to computer accessories, like the speakers used. To arrest the problem, the researcher promised the test subjects that better equipment would be used in the sessions to follow. The system interface was met with low tolerance levels. This was due to the use of the unfriendly command line interface. However, with more training and practice the testers eventually became familiar with the system environment.

5.5 Training of the System

This section of the Chapter explores much on the system training and trying to test how accuracy is the system to convert isiXhosa text to isiXhosa speech. The system was trained in such a way that one word or a sentence was repeatedly tested in 20 times to check how accurate the system is, when the word is being typed as input several times. Table 5-4 shows, the words that were used and the results that were obtained during the training of the system.

Table 5-4: Words Used for System Training

Data Used (Words)	Results
Molweni	20
Bhuti	17
Sisi	18
Qaqamba	17
Kunjani	16
Namhlanje	16
Igqirha	15
Ungonqongqothwane	15
Hambani	19
Igama	17
Imozulu	18
Abantwana	19
Ibhotile	15
Ninganxoli	14
Wagqibelanini	15
Namkelekile	14
Kudala	16
Ikrisimesi	18
Ulwimi	16
Ixesha	17

Table 5-4, shows the data that was used during the system training and the results obtained. The words were chosen randomly. The training of the system was repeated 20 times in one word. The main reason was to test or train the system whether it can pronounce the words repeatedly saying the something properly and clearly. According the results the word *Molweni* (hello in English) was pronounced 20 times correctly from first attempt. But we found there is a change of tone when it pronounces *bhuti* (brother), it pronounced the word up until the 17 times but the tone changed after that, we noted that it remove *h* from *bhuti*. The reason why is doing that was in phoneset there is *b* and *bh* consonants the system was a bit confused which one is right for the

word. Something happened to the *sisi* (my sister), sometimes it was putting *h* in front of *s* and pronounce the word *shishi*. The *qaqamba* was pronounced perfect except that the system sometimes instead of *qaqa* put *khakha*. The system sometimes it was confusing the consonants *nj* and *ny*, for example the words *kunjani* and *nyani* sometimes it is hard to hear the difference between this two words. All the words in the table were used during the system training are plotted in the chart. Figure 5-10, shows the chart for words which were used.

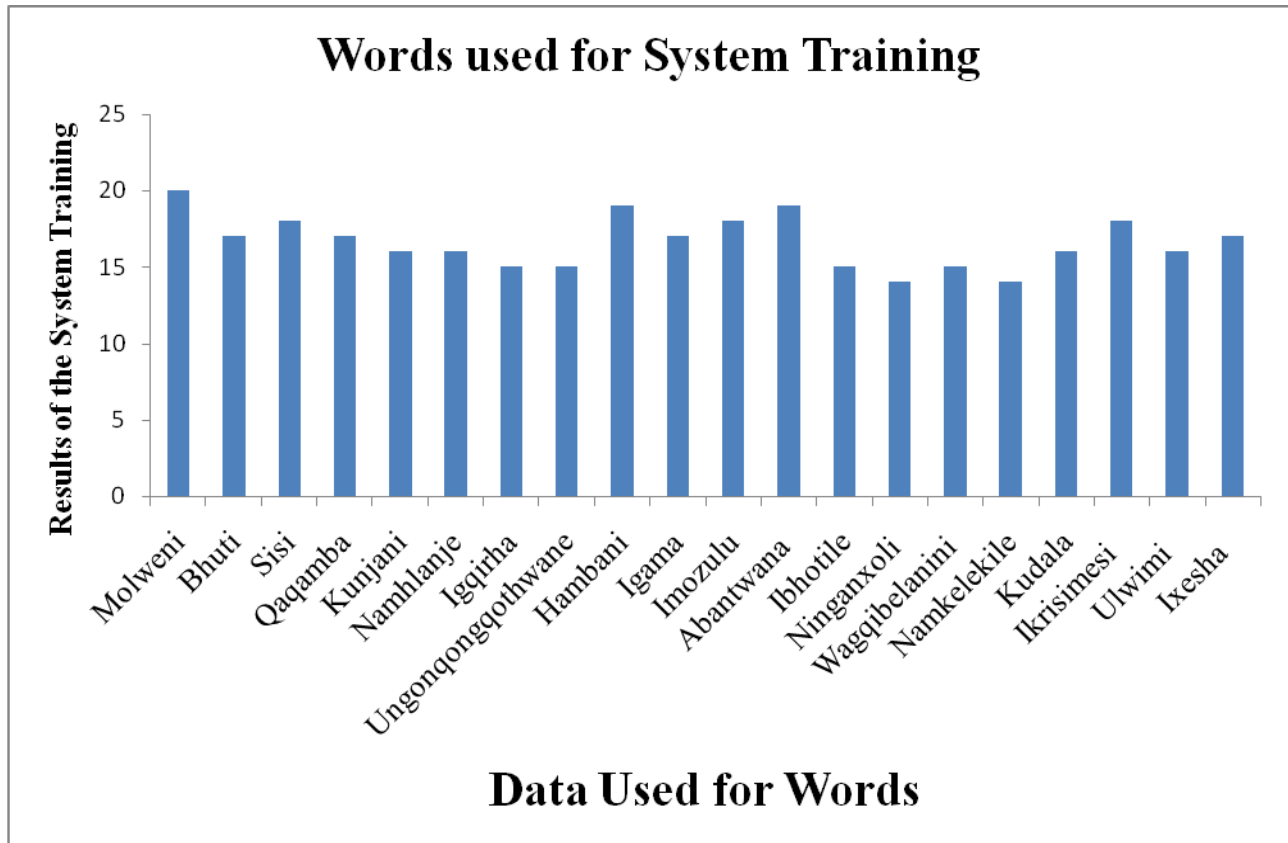


Figure 5-10: System Training Results for Words

The analysis of results from the training of the system shows that, 85 percent (%) of the words work perfect because the system was able to make a conversion of text to speech. The users were satisfied about how the system performs same of the conversion. Table 5-5; show the sentences that were used during training of the system.

Table 5-5: Sentences Used for System Training

Data Used (Sentence)	Results
Uyindoda emadodeni	19
Molweni baphulaphuli	20
Uphethwe yimincili	17
Nceda bhalela ekhaya	18
Ukuthatha inxaxheba	16
Ndifuna ukutya okunesondlo	15
Ndifuna ukukhwela uduladula	14
Kwiveki elandelayo ndahamba kunye naye apho ndathi ndahlangana	18
Yasimema ukuba sihambe kunye nayo xa isiya kukhonza	15
Yaxelela mna nenkosikazi yam ngalo mva ayo okuhamba kule nkonzo	17
Kulapho ke yazuza uxolo lomphfumlo	16
Abantu belo bandla bayikhongozela umfundisi wayithandazela nokuyithandazela	17
Kusenjalo mntu uthile wayimema ukuba ityelele ibandla labalindi yaya	15
Isiwa ivuka izama ukukhulisa abantwana abathathu iyodwa	14
Imana isiba namaphupha oyikisayo ebesuku ibhuqwa nasisithukuthezi	18
Umothuko endaba nawo esakufa wabangela ukuba idodobale impilo yam nentombi yam isentlungwini nayo	13
Ndingowohlanga lamaRomani ndiyinzalelwane yaseIndiya naxa ndhlala apha eYurophu	14
Thina amaRomani siluluntu olusondeleneyo ngakumbi iintsapho zethu	15
Uthe esakusweleka umkhwenyana wam kutshanje ndaphantsa ndafa	16
Wayeneminyaka engamashumi amathathu	18

The data in Table 5-5, shows the sentences which were chosen during the system training. The sentences were the mixture of long and short sentences. The main reason of choosing different sentences was to check whether the system can pronounce these two types using the same tone. According to Table 5-5, the results presented show different values meaning that the sentences have different level of phonetic descriptions. For instance, the first sentence *uyindoda yamadoda* (you are a man amongst men... in English), has a value of 19 which is totally different from the sentence which 13. It was easy for the system to pronounce the short sentences than long sentences that's why there is big difference between values in the results column. We noticed that the longer the sentence the more the system lost the mean of the sentence. There are some Xhosa songs that were used during system training. The system can read the words from the song without producing the lyrics of the song. All different angles of the system were trained and tested as it shown in these two tables of information. The results of the sentences were plotted in Figure 5-11.

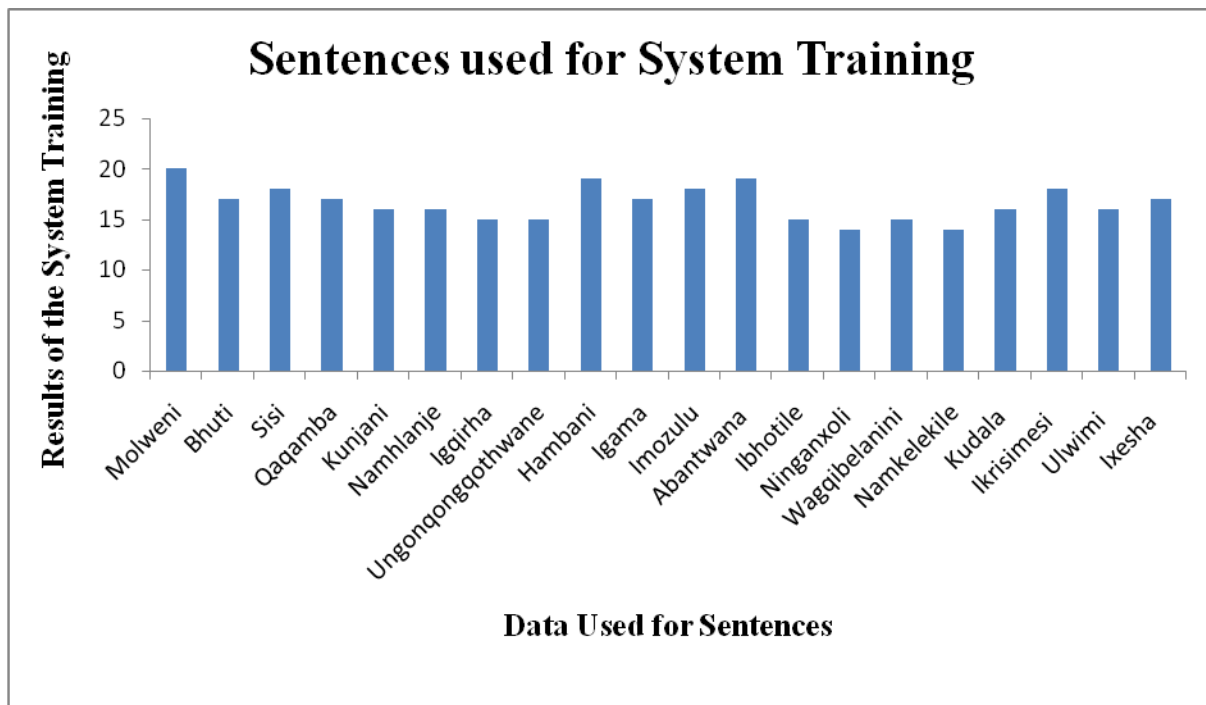


Figure 5-11: System Training Results for Sentences

Figure 5-11, shows the results that were obtained during the training of the system. Since the

system is able to perform the conversion of text to speech, we can conclude that the system can be used effectively in marginalized rural areas. The results from Table 5-10 and 5-11 were combined in one graph. Figure 5-12, shows the results from words and sentences data.

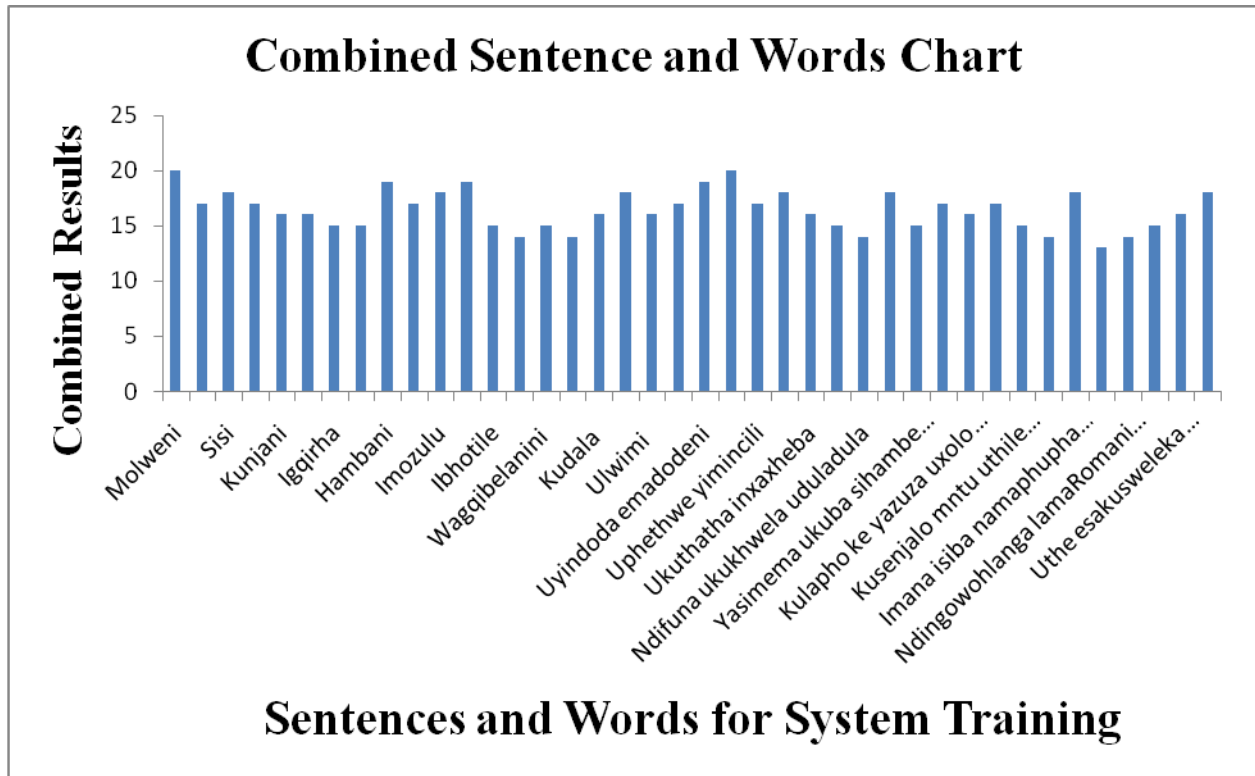


Figure 5-12: Combined Results of the System Training

Figure 5-12, shows the combined information about words and sentence that were used during the system training.

5.6 Conclusion

This chapter has demonstrated the methods followed in the performance of tests on the system. It discussed the functionality and usability tests that were conducted using certain metrics. The results are presented primarily in the form of graphs and tables. After each test was conducted, the results were discussed to address the objectives being measured. The next chapter presents a summary of the research. It will also discuss the conclusions which were arrived at, based on the goals and objectives of the research, before highlighting the challenges and outlining future work.

6 CHAPTER 6: SUMMARY, DISCUSSION AND CONCLUSION

6.1 Introduction

The main objective of this research is to develop and implement the Text-to-Speech system for use by people in a marginalized community, like Dwesa. The main driving factor is that not all users in the communities targeted for ICT4D deployment are able to make maximum use of the tools or services given to them. As previously highlighted, the platforms used when deploying these services are mostly presented in English, a language which the local inhabitants are not familiar with. The majority of these people use isiXhosa as their native language for communication purposes and, as such, it is difficult for them to use these ICT platforms. In this chapter, a summary of the research process is presented in Section 6.2, before a presentation of the achievements of the set objectives in Section 6.3. In Section 6.4, the challenges encountered are outlined. In Section 6.5, future work related to the extension of the system is presented and, in Section 6.6, the chapter is concluded with a discussion of the test results.

6.2 Summary of the Research

This research has developed and implemented a Text-to-Speech system for people living in rural areas of the Eastern Cape, specifically Dwesa. Open-source software called Festival speech synthesis was used to implement the system. The reasons why we chose open-source software include: free of cost, robust online community, support, portability, can easily be integrated into an existing platform and allows for code modification and reuse.

Community members, teachers and students were involved in the implementation of the system. The main reason for involving them was that the system was developed to meet their needs in the area because, in Dwesa, there are some people who cannot read and write in their mother tongue. During interviews with and the training of participants or users about the system, the community members, teachers and students were very happy about what the system was going to do. During the testing period, the participants (community members, teachers and students) showed their satisfaction regarding the system.

The research was exploring the development, implementation, testing and the deployment of the

text-to-speech system in the rural area of Dwesa. The Linux operating system was used as a platform during the implementation of the system. Therefore, this research presented the all steps that were taken in order to accomplish the intended results.

The motivation behind the implementation and development of the system that converted text to speech was based on the fact that rural communities of the Eastern Cape have a problem of the language barrier, in the use if ICT tools. The main reason for developing the Text-to-Speech system was to introduce illiterate people in the use of ICT tools.

It was previously mentioned that the Dwesa area consists of people who cannot write and read their mother tongue, which is isiXhosa. These people feel like they are not accommodated in the Siyakhula Living Lab project because all the systems which are already deployed in that area are written in English. People from Dwesa struggle to read their mother tongue, isiXhosa, and even more so, English. During the interview, illiterate people complain that even computers are written in English and they therefore cannot use them. The system was implemented to close the digital divide between community members. Those who know how to write or read help those who cannot do so by typing a simple sentence using the system, then the people listen.

6.3 Achievements of the objectives

The overall objective of the study was to develop and implement a Text-to-Speech system as a tool to help rural communities convert text-based web interfaces and information to sound. Based on the system requirements obtained through the use of interviews with network users, a Text-to-Speech system was developed and implemented. This objective had sub-objectives that have been addressed as follows:

Objective 1: To investigate the most preferred method of communication employed by rural communities with ICT tools, by conducting a literature review and through observation.

The first sub-objective was to investigate the ways in which people interact with ICT tools in marginalized communities. This was achieved through conducting a literature survey which was presented in Chapter 2 and interviews which were presented Chapter 3. It was also observed that most devices that are used in marginalized areas, to interact with ICT tools, are mobile phones

because users can make calls and send SMSs from them.

Objective 2: To investigate the ways in which people interact with ICT tools in marginalized communities, by performing a literature review and holding interviews.

The second sub-objective was to assess the ways in which illiterate people engage in communication. This objective was achieved through a literature survey and observation. The illiterate engaged in communication via their children. The children read news from the internet and then explain this to the illiterate. The other thing is that children help their illiterate elders by sending SMSs and making calls on their behalf because the illiterate cannot write an SMS. It was also noted, through observation, that children are the pillars of the lives of illiterate people; at times, they assist them in issues like saving important numbers to their phones.

Objective 3: To assess the ways in which illiterate people engage in communication, by conducting literature surveys and through observation.

The third sub-objective was to assess the way in which illiteracy affects the use ICT tools. Through the literature review, interviews, questionnaires and observation it was discovered that the lack of skills amongst illiterate people affects the use of ICT tools. ICT tools are deployed to improve the standard of living in rural areas, but it was observed that illiteracy affects the socio-economy of marginalized areas.

Objective 4: To investigate the extent to which language is a barrier to ICT use, by conducting informal interviews, performing a literature review and through observation.

The fourth sub-objective was to investigate the extent to which language is a barrier to ICT use. Through the literature review, interview and observation, it was discovered that if the language is different from the one commonly used in the area, there is a barrier to the use of the ICT tools deployed because people are not able to use the projects. For instance, in the case of Dwesa, the language that is used for communication is isiXhosa but the ICT tools are written in English. Most of the people in the area cannot access the information because of the language; this is why it is termed a language barrier in this project, because it has some limitations for some people.

Objective 5: To assess the ways in which illiterate people affect the use of ICT tools, by performing a literature review, conducting informal interviews, distributing questionnaires and through observation.

The fifth sub-objective was to assess the ways in which the Text-to-Speech system can improve the usability and uptake of ICT tools/services. Through the literature review, interviews and observation, it was observed that people were happy about the way in which the system is going to work. The system accommodates even people who cannot read and write isiXhosa and those who can read and write, but they do not understand English. The Text-to-Speech system will improve the use of ICTs because it allows people who have a language problem to perform the same tasks using ICT tools. Even during training and testing, people show satisfaction regarding the system performance.

Objective 6: To assess ways in which the isiXhosa TTS system improves the usability and uptake of ICT tools/services, by conducting informal interviews, performing a literature review and through observation.

The sixth sub-objective was to investigate the preferred method of communication in rural areas. Through the literature review and observation, this objective was achieved. It was observed that old people prefer face-to-face communication and communication via the radio and television. It was also observed that they prefer announcements that are usually made in funerals about the upcoming events. The youth prefer to use social networks such as facebook, 2go, twitter, mxit and watsup to communicate with others. In other observations, it was found that they also use the internet to communicate by sending emails.

Objective 7: To design and implement a software package for converting isiXhosa text into isiXhosa speech through prototyping and implementation.

The final sub-objective was to perform tests on the system, based on requirements. The main tests that were conducted were functionality, usability and the user acceptance analysis. The functionality test was performed with the use of various sets of isiXhosa words and sentences which were converted to sounds for use by Dwesa community members. Although the system interface was command driven which becomes a problem to novice users when it comes to the

use of many commands; the majority of users managed to familiarize themselves with it and performed some Text-To-Speech conversion. Tests were also conducted to gauge users' acceptance of the system through the questionnaire and the majority of users seemed to appreciate the initiative, as indicated by the results in the previous chapter.

6.4 Problems Encountered

There were many problems that we encountered during the implementation of the text-to-speech system. The first challenge was that during the introduction of the system to the participants (teachers, students and community members) they have no idea what the system is about. The researcher was forced to give a thorough explanation so that the participants could fully understand.

The second challenge was that there are some words on Festival open-source software which cannot be translated directly into isiXhosa. The commands in Festival speech synthesis are written in English so it is not easy to translate them, except if one builds new festival software from scratch. Community members were not comfortable with the language and the researcher tried to translate some of the commands.

The third challenge was that this research included two different fields of study: Science and Linguistics. We have a little bit of background about linguistic studies; this was a big challenge to the developer. These two fields of study were combined and the researcher came up with a system that could be used in both studies.

The fourth challenge was that during the training and testing of the system, it was found that participants were not attending the computer literacy classes. Even during the testing process, participants were not attending. Those who attended the training the first time were nowhere to be found the following day, but new people came for training. This consumed a lot of time because the researcher was forced to start afresh everyday, repeating the same thing.

The fifth challenge was to ensure that the Text-to-Speech system meets the needs of the Dwesa area. The researcher was supposed to develop a simple application that can be used in rural areas, without forgetting the kind of people in Dwesa. The illiterate people were very interested to

know more about the system and training was conducted amongst participants to show their satisfaction with the Text-to-Speech system.

The sixth challenge was that during the implementation of the system we first use the lexicon module but we experienced a lot of errors saying the isiXhosa lexicon is not defined. We find that the lexicon module is used for certain languages. It did not work for isiXhosa because there are a lot of clicks involved. The disadvantage of the lexicon module is that it only works with the words or sentences which are in the database; it also requires a large database. If the word is not in the database, the system fails to pronounce it. Ultimately, the Letter-to-Sound rule, which is the opposite of the lexicon module, was used.

The seventh challenge was that during the development and implementation of the system, the computer crashed several times because Festival is large open-source software. The other time grab (file on Linux operating system) was and the computer was formatted; we were forced to start from the implementation from scratch.

6.5 Future Work

There are a number of future extensions of the Text-to-Speech system which could accommodate the different needs of people in rural areas. It was mentioned in the discussion of the challenges encountered that participants did not clearly understand the work of the Text-to-Speech system because the system is new to the Dwesa network.

The future extensions that are planned are to come up with a manual that contains all the festival engine command, however, at this time are written using isiXhosa to accommodate people who are living in the rural areas of the Eastern Cape.

The Text-to-Speech system was developed to be used on computers; the future extension is to implement an application that can be available on mobile phones. So that the application can read SMSs that are written in isiXhosa, for people who cannot read and write.

There is no isiXhosa voice on Festival speech synthesis; the SLL project plans to extend the system by integrating the application on Festival so that it can be used all over the world.

Other possible future extensions are to work hand-in-hand with teachers who teach isiXhosa in schools, so that the system can be used in the learning environment. The students can upload an isiXhosa textbook on festival and run a few commands, then listen using a computer.

Blind people who use isiXhosa as their mother tongue where not accommodated in the applications that already exist. The SLL plans to extend the system so that it can accommodate such people.

6.6 Discussion of Results

The feedback about the training and testing, as plotted in Figure 5-9, shows that the implementation of the Text-to-Speech system for people who speak isiXhosa was very favorable in rural areas of Dwesa. The participants were very happy about the implementation and functionality of the system. The system was primarily supported by community members who said the system was going to help people who cannot write or read in the community. Others were not sure whether the system would make any difference in their living standard.

There was hope on the part of participants that, if the Text-To-Speech system is implemented properly and the time for training is enough, the system will be helpful in the community. The participants expressed their satisfaction with the content and functionality of the system because they can perform some tasks such as typing simple words such as *Molweni* (Hello) in the command line. The instructions were given using isiXhosa as the language of communication in the rural areas of Eastern Cape. It was easy for the participants to ask questions, provide comments, and discuss the system using their language because the researcher also uses the same language to communicate with them.

Since, in rural areas, there is a digital divide amongst members, the participants agreed that the system can bridge the gap. They expressed themselves by saying the system accommodates all community members. The successful operation of the system depends on the members of the community. Training and testing was done successfully, and the only thing that is left is to deploy the system and to ensure that the rural community members make use of it.

6.7 Conclusion

This thesis has described the design and implementation of a Text-to-Speech system to be used by rural Dwesa communities to convert isiXhosa words to sound. This work also demonstrated how the conversion of words and sentences is done in an exercise which was welcomed by the majority of test subjects during the testing process. Festival, an open-source software package, was used as the main system framework for the development of system databases and functions for the conversion process. This research also highlighted some of the needs of the community, in the area of communication, by motivating for an alternative means of communication which would encourage everyone across age groups to embrace ICTs. It was found that this system will not only help aged people but even the academically challenged, who are can neither understand English nor read textual information. The research development was based on open-source software so that it can be extended in future and can be integrated easily into the Dwesa network, because all the applications that are deployed there are written in open-source.

7 References

Abedjjeva E., Murray I., Arnott J. (1993). Applying Analysis of Human Emotion Speech to Enhance Synthetic Speech. Proceedings of Eurospeech 93 (2): 909-912.

Acacia. (2000). Information and communication technologies (ICTs) for improved service delivery in the new South Africa. [Last Accessed 15 September 2010] from <http://www.citizens.csir.co.za/>

Acero A. (1998). Source-Filter Models for Time-Scale Pitch-Scale Modification of Speech. Proceedings of ICASSP98.

Aitchison, J and Harley, A. (2001). South African illiteracy statistics and the case of the magically growing number of literacy and ABET learners.

Alam F, Nath K.P, and Khan M(2007). text to speech for Bangla language using festival, BRAC university, Bangladesh.

Bali, K (2004), Tools for the development of a Hindi speech synthesis system.

Balland C, Herreman D and Bell R (1998). Data Modeling Techniques for Data Warehousing. *International technical support organization SG24-2238-00,IBM redbook, 26 February 1998*

Barnard, E (2005). a general-purpose isiZulu Speech Synthesizer: Human Language Technologies Research group. Meraka Institute.

Bickley C, Syrdal A, and Schroeter J (1998). Speech Synthesis: in the Acoustics of Speech Communication, Picket J.M, Ed., Boston, NY:Allyn and Bacon.

Black (2000). Speech Synthesis in Festival: A practical course a making computer talk, edition 2.0, for Festival version 1.4.1.

Black A. W (2000). Flite: small run-time synthesizer: language technologies institute Carnegie Mellon University. [Online available: <http://cmuflite.org>]

Black, A Taylor, P and Caley, R, (1998). Edinburgh University, Center for speech technology

research, <http://www.cstr.ed.ac.uk/projects/festival/>, [last accessed March 2010]

Bosch, S (2009). An African language is the writing on the screen? Online Available at: http://www.merak.org.za/hlt_projects.htm [last Accessed: May 2010].

Breen A., Bowers E., Welsh W. (1996). An Investigation into the Generation of Mouth Shapes for a Talking Head. Proceedings of ICSLP 96 (4).

Chauhan A, Chauhan V, Singh G, Choudhary C and Arya P(2011). Design and development of a Text-To-Speech Synthesizer System:Journal, IJECT Vol.2, Issue 3, September, 2011.

Cooperation Framework on innovation Systems between Finland and South Africa (COFISA). (2008). Using ICTs to Optimise Rural Development. Available at: www.cofisa.org.za [Last accessed: October 2010]

Costello, J.B. (2000). Education: The fuel for tech's Golden Age. Electronic Business. [Last accessed July 2010] from <http://www.e-insite.net/eb-mag/index.asp?layout=article&articleId=CA53574&stt=001>

Curtain, R. (2003). Information and Communications Technologies and Development: Help or Hindrance? Kamran Jebreili Associated Press.

DALVIT, L., R. ALFONSI, N. ISABIRYE, S. MURRAY, A. TERZOLI AND M. THINYANE (2006). A case study on the teaching of computer training in a rural area in South Africa. 22nd CESE(Comparative Education Society of Europe) conference, Granada, Spain, Department of Education (DoE) and Department of Communication (DoC) (2001). Strategy for Information and Communication Technology in Education, Government Printer, Pretoria.DWESA PROJECT. (Accessed 25 June 2010): <http://www.dwesa.org>

Dalvit, L., Thinyane, M., Muyingi, H. and Terzoli, A. (2007). The Deployment of an e-Commerce Platform and Related Projects in a Rural Area in South Africa. International Journal of Computing and ICT Research, ISSN 1818-1139, Vol.1, NO.1, pp.9-18, June 2007

Ding, F. (2006). Modular design for Mandarin Text-To-Speech Synthesis.

Dyakalashé S (2009). Cultural and linguistic localization of the virtual shop-owner interface of the e-Commerce platform for rural development.

Dyakalashé S, Terzoli A. and H.N Muyingi. (2009). Cultural and Linguistic Localization of the Virtual Shop-Owner Interfaces of E-Commerce Platforms for Rural Development.

Elisha J. Martha (2006). The Application of Information and Communication Technology (ICT) in Nigerian Academic Libraries prospects and problems. *The Information Manager Vol.6 (1 & 2) 2006*

Festival speech synthesis system – 24 voices.

Furlonger, D. (2002, January 25.) Rally to read. [Last accessed 4 October 2010] from <http://free.financialmail.co.za/rallytoread/rally.htm>

Gakuru M and Ngugi K (2005). Development of a Kiswahili text-to-speech system, University of Nairobi.

Gaved M. (1993). Pronunciation and Text Normalization in Applied Text-to-Speech Systems. *Proceedings of Eurospeech 93 (2): 897-900.*

Goudge (2007) Coping with the cost burdens of illness: Combing qualitative and quantitative methods in longitudinal, household research. *Scand J Public Health 2007 35: 181* the online version of this article can be found at: http://sjp.sagepub.com/content/35/69_suppl/1818 [last accessed: 20 October 2010]

Hakulinen J. (1998). Suomenkieliset puhesynteesiohjelmistot (The Software Based Speech Synthesizers for Finnish). Report Draft, University of Tampere, Department of Computing Science, Speech Interfaces, 26.8.1998

Hallahan W. (1996). DECTalk Software: Text-to-Speech Technology and Implementation. *Digital Technical Journal.*

Herselman, M.E. (2003). ICT in Rural Areas in South Africa: Various Case Studies. *Technikon Pretoria, South Africa. Informing Science Proceedings, Citeseer.*

Hess W. (1992). Speech Synthesis - A Solved Problem? Proceedings of EUSIPCO 92 (1): 37-46.

Hlungulu, B and Thinyane, M (2010). Building an e-health component for a multipurpose communication centre for a marginalized community, using FOSS.

Hon H., Acero A., Huang X., Liu J., Plumpe M. (1998). Automatic Generation of Synthesis Units for Trainable Text-to-Speech Systems. Proceedings of ICASSP 98 (CD-ROM)

Hood, M (2004). Creating a Voice for Festival speech Synthesis system.

http://www.cstr.ed.ac.uk/projects/festival/manual/festival_24.html [Accessed:26/03/2010]

Huang X., Acero A., Adcock J., Hon H., Goldsmith J., Liu J., Plumpe M. (1996). Whistler: A Trainable Text-to-Speech System. Proceedings of ICSLP96 (4)

Huang X., Acero A., Hon H., Ju Y., Liu J., Mederith S., Plumpe M. (1997). Recent Improvements on Microsoft's Trainable Text-to-Speech System - Whistler. Proceedings of ICASSP97 (2): 959-934.

Jakachira B.T, Terzoli A. and H.N Muyingi (2009) Implementing integrated e-Government functionality for a marginalized community in the Eastern Cape, South Africa.

James, T. (2001). An Information Policy Handbook for Southern Africa. CD ROM. IDRC

Jere R, Nobert. (2009) Implementation of a rewards-based negotiation module for an e-Commerce platform.

Jere, N. (2009). Implementation of a rewards-based negotiation module for an e-Commerce platform, South Africa. M.Sc. Thesis. University of Fort Hare.

Juang B.H and Lawrence R. Rabiner (2004). "Automatic Speech Recognition – A brief history of the technology development

Kendall, S(2001). Constraints to growth and employment in South Africa: report No.2: evidence from the small, medium and micro enterprise firm survey.

- Klatt D. (1987). Review of Text-to-Speech Conversion for English. Journal of the Acoustical Society of America, JASA vol. 82 (3), pp.737-793.
- Kunjuzwa, D.T and Thinyane, M. (2009). Exploring User-Driven Telephony Services in an Information and Communication Technology for Development Context.
- Louis Fourie Consultants (2008). ICT for rural livelihood South Africa: Available at:<http://www.ict4rl.info/Country/SouthAfrica> [Last accessed: August 2010]
- Makombe, F, Thinyane, M. and Terzoli, A (2011). Developing a help-desk system for a multi-purpose ICT platform in a marginalized setting.
- Merson Paulo. (2009). Data Model as an Architectural View.
- Morton, K. (1987). The British Telecom Research Text-to-Speech Synthesis System - 1984-1986. Speech Production and Synthesis. Unpublished PhD Thesis. University of Essex. pp. 142-172.
- Moyo T, Thinyane, M and P Kogeda (2010). Reengineering Legacy Applications for SOA Middleware Integration.
- Murray I., Arnott J., Alm N., Newell A. (1991). A Communication System for the Disabled with Emotional Synthetic Speech Produced by Rule. Proceedings of Eurospeech 91 (1): 311-314.
- Naidoo, S. (2002). Education in rural schools. [Last accessed 4 June 2010] from <http://www.mcetail.co.za/corporate/rallytoread/Background/rural.html>
- Ngwenya S, Terzoli A, Thinyane, M and Gumbo(2010). Developing a Context-Sensitive Revenue Management System for ICT4D projects in Rural Marginalized Communities.
- Odasz, F. (2004) Sustainable Ecommerce Entrepreneurship Development Strategies: A Rural Community Future-Proofing Program. Lone Eagle Consulting.
- Pade, C.I. (2008). The development and implementation of an evaluation framework for rural ICT projects in developing countries: exploring the Siyakhula Living Lab in South Africa.

Palmer R., H. Timmermans and D. Fay (2002), From conflict to negotiation: nature-based development on South Africa's Wild Coast. Pretoria: Human Sciences Research Council; Grahamstown: Institute of Social & Economic Research, Rhodes University.

Parssinen, K. (2007). Multilingual Text-To-Speech System for mobile devices: Development and Applications.

Portele T., Höfer F., Hess W. (1994). A Mixed Inventory Structure for German Concatenative Synthesis. University of Bonn.

Portele T., Krämer J. (1996). Adapting a TTS System to a Reading Machine for the Blind. Proceedings of ICSLP 96 (1).

Portele T., Steffan B., Preuss R., Hess W. (1991). German Text-to-Speech Synthesis by Concatenation of Non-Parametric Units. Proceedings of Eurospeech 91 (1): 317-320.

Portele T., Steffan B., Preuss R., Sendlmeier W., Hess W. (1992). HADIFIX - A Speech Synthesis System for German. Proceedings of ICSLP 92 (2): 1227-1230.

Prinsloo M. (1999). Literacy in South Africa.

Rousseau, F. and Mashao, D. (2004). Increased Diphone Recognition for Afrikaans Text-To-Speech.

SALT J. (1992). The Future of International Labour Migration, International Migration Review, Vol. 26, No. 4. pp. 1077-1111.

Samalenge J. and Thinyane M (2010). Developing SOA Wrappers for Communication Purposes, in Rural Areas.

Schroeter, J. (1996). Text to Speech Synthesis.

Scott M.S, N. H. Muyingi, Prof. A. Terzoli and Dr. M. Thinyane (2010). Investigation and Development of an e-Judiciary Service for a Citizen-Oriented Judiciary System for Rural Communities.

Siyakhula project. (2008). Dwesa e-Commerce platform project: Available at: www.dwesa.org

Statistics South Africa. (2006). Available at: <http://www.statssa.gov.za/publications/Report-03-04-02/Report-03-04-02.pdf> [Last accessed September 2010]

Summer Session Beyond Kyoto. (2000). Achieving Sustainable Development. Hamburg.

Takara, T. and Anberbir, T. (2006) Development of an Amharic Text To Speech System Using Cepstral Method.

Tarwireyi, P., Terzoli, A. and Muyingi, H. (2008). Adapter-based revenue management system for the exploration of non-conventional billing options in new markets for telecommunications. SATNAC conference Wild Coast, Eastern Cape Province, South Africa. Available at: www.satnac.org.za/proceedings/2008/management.htm [Last accessed: May 2010]

Thinyane H (2009). Sim or application layer? An implementation level Analysis on the use of mobile phone for ICD development.

Timmermans, H.G. (2004). Rural livelihoods at Dwesa/Cwebe: Poverty, development and natural resource use on the Wild Coast, South Africa. M.Sc. Thesis. Rhodes University

Ungana Afrika. (2007). Rural ICT Support and Development. Available at: http://www.ungana-afrika.org/projects/concept_paper_v1.1.pdf [Last accessed: June 2010]

Vile S David, Olesti C David and Pallares. (2009). Adaptation of voice server to automotive environment.

Vosloo, S. (2003). Best Practices of Information and Communication Technology for Development (ICT4D) Projects.

Waters K., Levergood T. (1993). DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces. DEC Technical Report Series, Cambridge Research Laboratory, CRL 93/4.

Wee C. Mee and Bakar A. Zaitun (2006). The obstacles towards the use of ICT tools in teaching and learning Information Systems in the Malaysian universities. *The international Arab Journal of Information Technology*, Vol.3 No.3, July 2006.

Weigal et al (2004). ICT4D – Connecting people for a better world.

Wertlen R.R. (2007). An Overview of ICT Innovation for Development Projects in Marginalized Rural Areas.

Yvon F, Boula de Mareuil P and Alessandro C.D. (1998). Objective evaluation of grapheme to phoneme conversion for Text-to-Speech synthesis in French. *Computer speech and language* (1998) 12, 393-410 Article No.1a980104

8 Appendix A – Technologies Required

This section explores the installation of the system components which are included as part of the research.

8.1 Operating System

The system was developed under the Linux operating system but it can also be developed under Windows. Since the project is in the context of Siyakhula Living Lab, the Linux operating system was the best choice because all the other projects under the SLL are developed under Linux. The Linux operating system version was described as:

Distributor ID : Ubuntu
Description : Ubuntu 10.04.3 LTS
Release : 10.04
Codename : Lucid

The Linux operating system is free and easy to obtain from the internet (<http://www.ubuntu.com>). This operation system was installed using a Disk. Linux come out with a full package of software such as Microsoft office.

Festival installation

Festival speech synthesis is free, open-source software that was used in the development of the system. The installation involves the following line of command shell.

Sudo apt-get install festival

This command shell installs all the packages related to festival.

Festival can be installed manually without the use of a command shell. Festival installed using a Synaptic Package Manager (SPM). The SPM provides the same features as **apt-get**. To open SPM go to system -> Administrator -> Synaptic Package Manager. Enter the administrator password then the page will appear. The SPM allows you to search Festival. After Festival, click mark for installation then apply.

There are other packages in Festival which are very important in the development of a new voice such as Festvox, festival-doc, which are installed using the command shell.

Sudo apt-get install Festvox -kallpc8k, which provides American English. Those packages can also be installed manually using the same steps on festival installation.

8.2 Gcc

Gcc is a compiler for C and C++ in Ubuntu Linux operating system. The gcc compiler was installed as follows: before the installation we need to update the system using the following command.

Sudo apt-get update, after the update, the upgrade was also required. The following command was used

Sudo apt-get upgrade

After the update and upgrade was completed, the gcc compiler was installed using the following command:

Sudo apt-get install build-essential

The gcc compiler can be installed manually using the same procedure explained in 7.2.

9 Appendix B – System Implementation

During implementation, isiXhosa Text-to-Speech modules were developed.

9.1 Phoneset Module

The following list shows how the phoneset module was implemented for the isiXhosa Text-to-Speech system.

```
(defPhoneSet
  ufh_isiXhosa_siphe_diphone
  ;; Phone Features
  (
    ;; vowel, consonant, diacritic, silence, closure or other
    (vc + - d s cl 0)
    ;; vowel length: shrt long diphtong schwa
    (vlng s l d a 0)
    ;; vowel height: high mid low
    (vheight 1 2 3 0)
    ;; vowel frontness: front mid back
    (vfront 1 2 3 0)
    ;; lip rounding
    (vrnd + - 0)
    ;; consonant types : [p] stop (plosives), [f] fricative, [h] affricate,
    ;; [a(l/r/g/o)] approximant(lateral/retroflex/gluide/other), [n] nasal,
    ;; [c] clicks, [t] tap/flap, [r] trill, [v] voiced implosives,
    ;; [sa] stop with aspiration, [ca] click with aspiration, [la] "lateral affricate",
    ;; [nc] nasalized clicks
    (ctype s f a al ar ag ao n c t r v sa ca la nc 0)
    ;; place of articulation: [b] bilabial, [i] interdental, [a] alveolar, [z] alveopalatal,
    ;; [p] bilabial, [l] alveolar, [d] dental, [v] velar,
```

```

;;          [g] glottal, [x] pharyngeal, [u] uvular,
;;          [r] retroflex
(cplace b i a z p l d v g x u r 0)
;; consonant voicing
(cvox + - 0)
)
;; Phone set members
(
;; (ph vc vl vh vf vr ct cp cv)
;; CONSONANTS (pulmonic)
;; Stops or Plosives
(p   - 0 0 0 0 0 s b -)      ; IPA 112
(b   - 0 0 0 0 0 s b +)      ; IPA 98
(t   - 0 0 0 0 0 s a -)      ; IPA 116
(d   - 0 0 0 0 0 s a +)      ; IPA 100
(k   - 0 0 0 0 0 s v -)      ; IPA 107
(g   - 0 0 0 0 0 s v +)      ; IPA 103
;; Stops with aspiration
(P   - 0 0 0 0 0 sa b -)      ; IPA 112
(B   - 0 0 0 0 0 sa b +)      ; IPA 98
(T   - 0 0 0 0 0 sa a -)      ; IPA 116
(K   - 0 0 0 0 0 sa v -)      ; IPA 107
(kh  - 0 0 0 0 0 sa v +)      ; HACK !!! "soft" k: voiced implosive?
;; Nasals
(m   - 0 0 0 0 0 n b +)      ; IPA 109
(n   - 0 0 0 0 0 n a +)      ; IPA 110
(ng  - 0 0 0 0 0 n v +)      ; ng
(ny  - 0 0 0 0 0 n p +)      ; IPA 110
;; Tril
(r   - 0 0 0 0 0 t a +)      ; IPA 114
(tr  - 0 0 0 0 0 t a +)

```


(l - 0 0 0 0 t a +)

(y - 0 0 0 0 t a +)

(w - 0 0 0 0 t a +)

:: Fricative

(f - 0 0 0 0 f l -) ; IPA 102

(v - 0 0 0 0 f l +) ; IPA 118

(s - 0 0 0 0 f a -) ; IPA 115

(z - 0 0 0 0 f a +) ; IPA 122

(g - 0 0 0 0 f v -) ; IPA 120

(h - 0 0 0 0 f g -) ; IPA 104

(hl - 0 0 0 0 f g +) ; HACK!!! actually breathy

(sh - 0 0 0 0 f p -) ;

(th - 0 0 0 0 f t +)

(rh - 0 0 0 0 f a +)

(dl - 0 0 0 0 f a +)

:: Lateral

(hl - 0 0 0 0 al a -) ; HACK!!! Lateral fricative

(s - 0 0 0 0 al a +) ; IPA 108 ; Lateral approximants

:: Glide

(j - 0 0 0 0 ag p +) ; IPA 106

(w - 0 0 0 0 ag v +) ; IPA 119

:: CONSONANTS (non-pulmonic)

:: Clicks

(q - 0 0 0 0 c d -) ; IPA 45 \

(c - 0 0 0 0 c a -) ; IPA 60 !\

(x - 0 0 0 0 c p -) ; IPA 248 =\

(qh - 0 0 0 0 c d +)

(nc - 0 0 0 0 c a +)

(xh - 0 0 0 0 c p +)

:: Clicks with aspiration

(Q - 0 0 0 0 ca d -) ; IPA 45 \

```

(C   - 0 0 0 0 ca a -)      ; IPA 60          !\
(X   - 0 0 0 0 ca p -)      ; IPA 248         =\
;; VOWELS
(i   + 1 1 1 - 0 0 0)       ; IPA 121         y
(u   + s 3 1 + 0 0 0)       ; IPA 232         }
;;(E  + 1 2 2 - 0 0 0)       ; IPA ???         }
;;(O  + s 2 3 + 0 0 0)
(e   + 1 1 2 - 0 0 0)       ; IPA 111
(o   + s 2 2 + 0 0 0)       ; ; IPA 207         {
(a   + 1 3 3 - 0 0 0)       ; IPA 191         V
(en  + 1 1 2 - 0 0 0)
;Affricate
(tsh - 0 0 0 0 a p -)       ; IPA 116 + IPA 83 (t + sh)
(ts  - 0 0 0 0 a a -)       ; IPA 100 + IPA 90 (t + s)
;;(dz - 0 0 0 0 a a +)       ; (d + z)
(nj  - 0 0 0 0 a p +)       ;
;"Lateral Affricate" - A hack!!!
(l   - 0 0 0 0 la a -)      ; IPA 241
(dl  - 0 0 0 0 la p +)      ; IPA 116 + IPA 83 (d + zh)
;; SILENCES
(pau s 0 0 0 0 0 0 0)       ; Silence
(#   s 0 0 0 0 0 0 0)       ; Silence
(H#  s 0 0 0 0 0 0 0)       ; Silence
(cl  cl 0 0 0 0 0 0 0)       ; Closure
(gs  s 0 0 0 0 0 0 0)       ; Glottal stop
)
)
(PhoneSet.silences '(pau # H#))
(define (ufh_isiXhosa_siphe_diphone::select_phoneset)
  "(ufh_isiXhosa_siphe_diphone::select_phoneset)
Set up phone set for ufh_isiXhosa_siphe_diphone."

```

```

(Parameter.set 'PhoneSet 'ufh_isiXhosa_siphe_diphone)
(PhoneSet.select 'ufh_isiXhosa_siphe_diphone)
)
(define (ufh_isiXhosa_siphe_diphone::reset_phoneset)
  "(ufh_isiXhosa_siphe_diphone::reset_phoneset)
Reset phone set for ufh_isiXhosa_siphe_diphone."
  t
)
(provide 'ufh_isiXhosa_siphe_diphone_phoneset)

```

9.2 Letter-to-Sound Rule

The following list shows how the letters or characters were linked using the letter-to-sound rule. This module was implemented and is helpful when the word is not in a database letter-to-sound rule links the letter and comes up with a string of words.

```

(Its.ruleset
ufh_isiXhosa_siphe_diphone
( (Vowel a e i o u) )
(
;; LTS rules
([a]=a)
([e]=e)
([i]=i)
([o]=o)
([u]=u)
([ "" a ] = a )
([ "" e ] = e )
([ "" i ] = i )
([ "" o ] = o )
([ "" u ] = u )
([ "" m ] = mm )
([ "" n ] = nn )

```

(["-" a] = a)
(["-" e] = e)
(["-" i] = i)
(["-" o] = o)
(["-" u] = u)
([b]=b)
([c h] = ch)
([d]=d)
([f]=f)
([g]=g)
([h]=h)
([h l] = hl)
([j]=j)
([k]=k)
([k h] = kh)
([l]=l)
([m]=m)
([n]=n)
([p]=p)
([p h] = ph)
([q]=q)
([q h] = qh)
([r]=r)
([s]=s)
([s h] = sh)
([t]=t)
([t h] = th)
([t l] = tl)
([t s h] = tsh)
([t s ""] = tsh)
([w]=w)

([y]=y)
(["- b] = b)
(["- c h] = ch)
(["- d] = d)
(["- f] = f)
(["- g] = g)
(["- h] = h)
(["- h l] = hl)
(["- j] = j)
(["- k] = k)
(["- k h] = kh)
(["- l] = l)
(["- m] = m)
(["- n] = n)
(["- p] = p)
(["- p h] = ph)
(["- q] = q)
(["- q h] = qh)
(["- r] = r)
(["- s] = s)
(["- s h] = sh)
(["- t] = t)
(["- t h] = th)
(["- t l] = tl)
(["- t s h] = tsh)
(["- t s ""] = tsh)
(["- w] = w)
(["- y] = y)
)

9.3 Phrasing Module

After the letter-to-sound module was implemented, the phrasing module was also implemented for the isiXhosa language. The following list shows how the phrasing module was implemented:

```
(set_backtrace t)
(define (token_next_punc token)
  "(token_next_punc token)
  Find next punctuation after the word"
  (if (null token) "."
      (let (res (punc (item.feat token "punc")))
        (if (string-equal punc "0") (token_next_punc (item.next token))
            punc))))
(set! ufh_isiXhosa_siphe_diphone_phrase_cart_tree
  '
  ((lisp_token_end_punc in ("?" "." "!" ))
   ((BB))
   ((lisp_token_end_punc in (";" ":" "-" "--"))
    ((B))
    ((lisp_token_end_punc in (","))
     ((B))
     ((n.name is 0) ;; end of utterance
      ((BB))
      ((NB))))))
(define (ufh_isiXhosa_siphe_diphone::select_phrasing)
  "(ufh_isiXhosa_siphe_diphone::select_phrasing)
  Set up the phrasing module for isiXhosa language."
  (set! phrase_cart_tree ufh_isiXhosa_siphe_diphone_phrase_cart_tree)
  (Parameter.set 'Phrase_Method 'cart_tree)
  (Param.set 'Phrasify_Method Classic_Phrasify)
  )
( define (ufh_isiXhosa_siphe_diphone::reset_phrasing)
```

```
"(ufh_isiXhosa_siphe_diphone::reset_phrasing)
Reset phrasing information."
```

```
t
)
```

```
(provide 'ufh_isiXhosa_siphe_diphone_phrasing)
```

Intonation Module

This module was implemented to determine the tone of the isiXhosa language. The following list shows how the intonation module was implemented for rural areas.

```
(set! ufh_isiXhosa_siphe_diphone_accent_cart_tree
'
```

```
(
(R:SylStructure.parent.gpos is content)
```

```
( (stress is 1)
```

```
((Accented))
```

```
((NONE))
```

```
)))
```

```
(define (ufh_isiXhosa_siphe_diphone::select_intonation)
```

```
"(ufh_isiXhosa_siphe_diphone::select_intonation)
```

```
Set up intonation for isiXhosa."
```

```
(set! int_accent_cart_tree ufh_isiXhosa_siphe_diphone_accent_cart_tree)
```

```
(Parameter.set 'Int_Target_Method 'Simple)
```

```
)
```

```
(define (ufh_isiXhosa_siphe_diphone::reset_intonation)
```

```
"(ufh_isiXhosa_siphe_diphone::reset_intonation)
```

```
Reset intonation information."
```

```
t
)
```

```
(provide 'ufh_isiXhosa_siphe_diphone_intonation)
```

10 Appendix C – Database

10.1 Words Table

A database which contains isiXhosa words was created. The following shows how the words database was created

```
("a" nil (( a ) 0 ))
("aa" nil (( a ) 1 ) (( a ) 0 ))
("aba" nil (( a ) 1 ) (( b a ) 0 ))
("ababaleka" nil (( a ) 0 ) (( b a ) 0 ) (( b a ) 0 ) (( l e ) 1 ) (( k a ) 0 ))
("abakhonzi" nil (( a ) 0 ) (( b a ) 0 ) (( kh ) 0 ) (( o ) 1 ) (( n z ) 0 ) (( i ) 0 ))
("abanenxaxheba" nil (( a ) 0 ) (( b a ) 1 ) (( n e ) 0 ) (( nx ) 1 ) (( a ) 0 ) (( xh ) 1 ) (( e ) 1 ) (( b a ) 0 )
))
("ababetya" nil (( a ) 0 ) (( b a ) 0 ) (( b e ) 1 ) (( ty a ) 0 ))
("abathandekayo" nil (( a ) 0 ) (( b a ) 0 ) (( th a ) 0 ) (( n d e ) 1 ) (( k a ) 0 ) (( y o ) 0 ))
("ababenayo" nil (( a ) 0 ) (( b a ) 0 ) (( b e ) 0 ) (( n a ) 1 ) (( j o ) 0 ))
("ababezama" nil (( a ) 0 ) (( b a ) 0 ) (( b e ) 0 ) (( z a ) 1 ) (( m a ) 0 ))
("ababhangqa" nil (( a ) 0 ) (( b a ) 0 ) (( B a ) 1 ) (( n a ) 0 ))
("ababhona" nil (( a ) 0 ) (( b a ) 0 ) (( B o ) 1 ) (( n a ) 0 ))
("ababiyelwe" nil (( a ) 0 ) (( b a ) 0 ) (( b i ) 0 ) (( j e ) 1 ) (( l w e ) 0 ))
("ababizwa" nil (( a ) 0 ) (( b a ) 0 ) (( b i ) 1 ) (( z w a ) 0 ))
```

10.2 Sentence Table

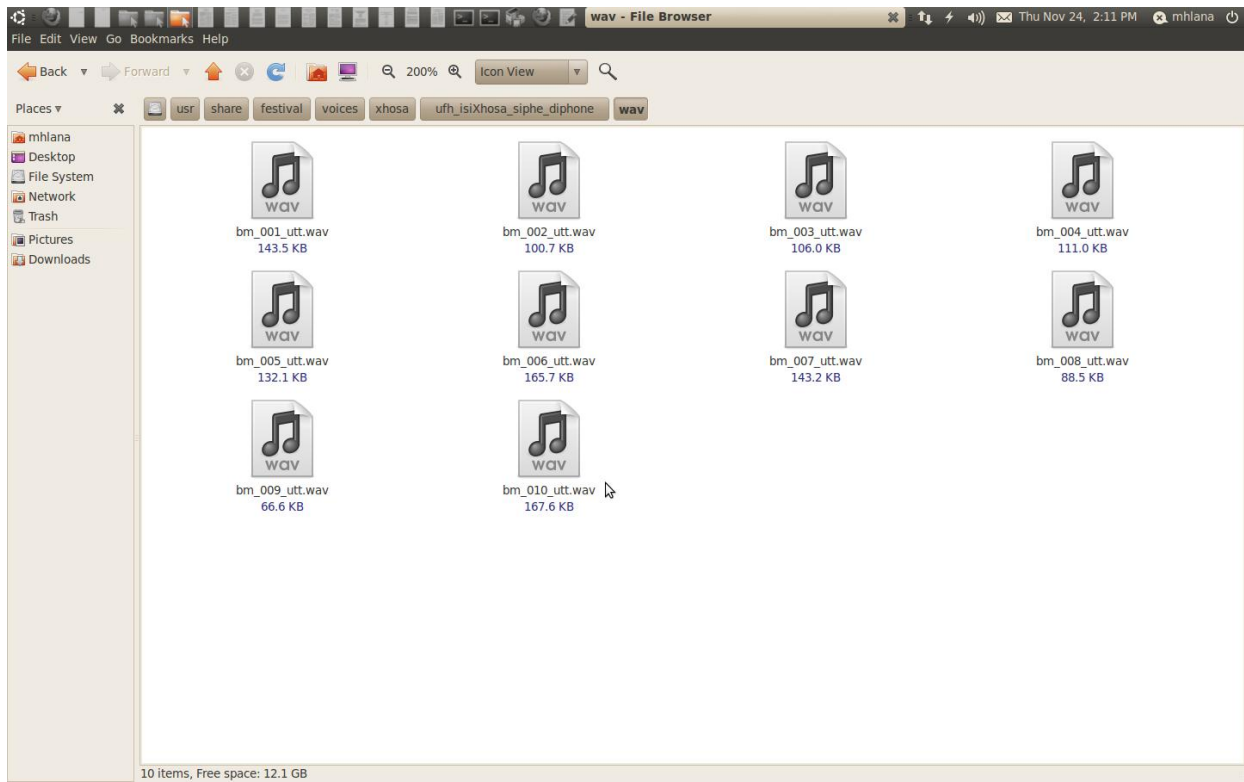
A database which contains isiXhosa sentences was also created for the isiXhosa text-to-speech system. The following shows how the database was created.

```
( bm_001_utt "ndingowohlanga lamaRomani ndiyinzalelwane yaseIndiya naxa ndhlala apha eYurophu")
( bm_002_utt "thina amaRomani siluluntu olusondeleneyo ngakumbi iintsapho zethu")
```


(bm_003_utt "uthe esakusweleka umkhwenyana wam kutshanje ndaphantsa ndafa")
(bm_004_utt "wayeneminyaka e-30 kuphela ndimthanda ngkungathi ungunyana wam ngqo")
(bm_005_utt "umothuko endaba nawo esakufa wabangela ukuba idodobale impilo yam nentombi yam isentlungwini nayo")
(bm_006_utt "imana isiba namaphupha oyikisayo ebesuku ibhuqwa nasisithukuthezi isiwa ivuka izama ukukhulisa abantwana abathathu iyodwa")
(bm_007_utt "kusenjalo mntu uthile wayimema ukuba ityelele ibandla labalindi yaya")
(bm_008_utt "abantu belo bandla bayikhongozela umfundisi wayithandazela nokuyithandazela")
(bm_009_utt "kulapho ke yazuza uxolo lomphfumlo")
(bm_010_utt "yaxelela mna nenkosikazi yam ngalo mva ayo okuhamba kule nkonzo yasimema ukuba sihambe kunye nayo xa isiya kukhonza")
(bm_011_utt "kwiveki elandelayo ndahamba kunye naye apho ndathi ndahlangana khona nomfundisi ongaqeqeshwanga ogama linguGeorge.")
(bm_012_utt "uGeorge wandixelela ukuba uyesu unokusinceda sijongane neentlekele ezisivelelayo endleleni yethu")
(bm_013_utt "ndathabatheka kakhulu yindlela azinikele ngayo nayindlela abayinceda ngayo intombi yam ukuba ifumanane nothixo")
(bm_014_utt "kangangokuba ndagqiba kwelokuba khe ndmnike ithuba ebomini bam uthixo")

10.3 Voice database

Voice database was created and linked to these two databases. The following screenshot shows how the voice database was created and connected to the other databases.



11 Appendix D – Usability and Functionality Testing

11.1 Usability testing – Questionnaires Part 1

After training, users were given a task to complete and a questionnaire was given to users. The questionnaire was based on testing the level of understanding, of the users, about the use of computers with internet and the age range of people who are illiterate.

Participants' background

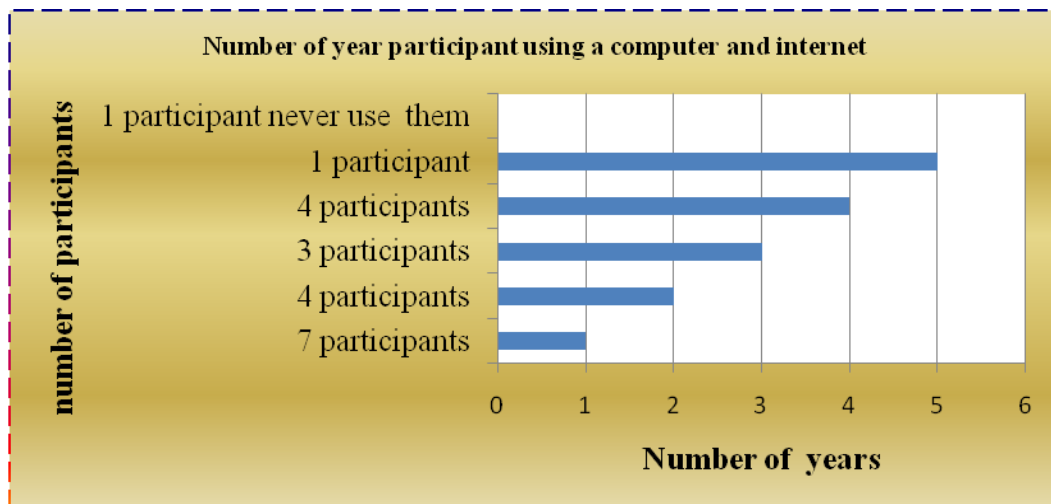
The section provides all the questions that were asked during testing.

Have you used a computer before?

The answer from participants was yes because they used a computer before.

How many years have you been using a computer and the internet?

The following chart shows the answers from participants.

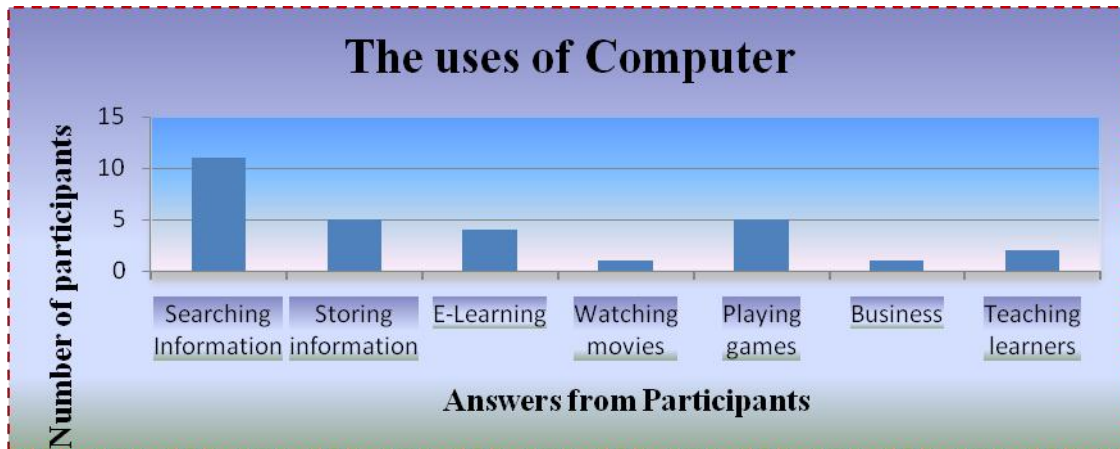


The chart shows that seven participants used a computer for a year.

What do you use a computer for?

The following chart shows the answers from the participants.

The participants used the computer for different purposes, as shown in the chart below.

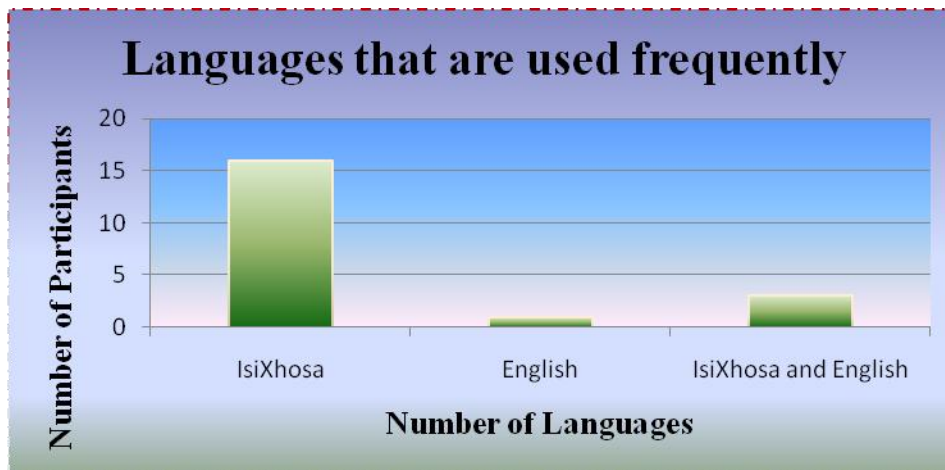


What is your home language?

Since the researcher is working with the rural community, Dwesa, the answer was that isiXhosa is their mother tongue or home language.

Which language do you use frequently?

The following chart shows the answers from participants.



Sixteen participants use isiXhosa frequently as their language; three participants use both isiXhosa and English, and one uses English. The participants who use English frequently are, mostly, teachers who use the language when they communicate with students in class.

Is the language chosen above the one you use to communicate with other community members?

The answer was yes.

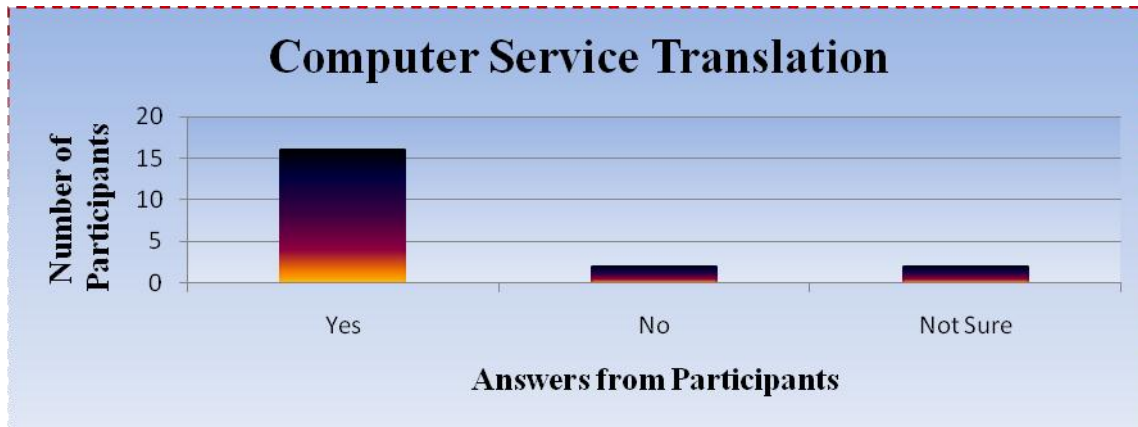
Do you think that if computers and the internet can use your own language it would make it

easier to use?

The answer was yes

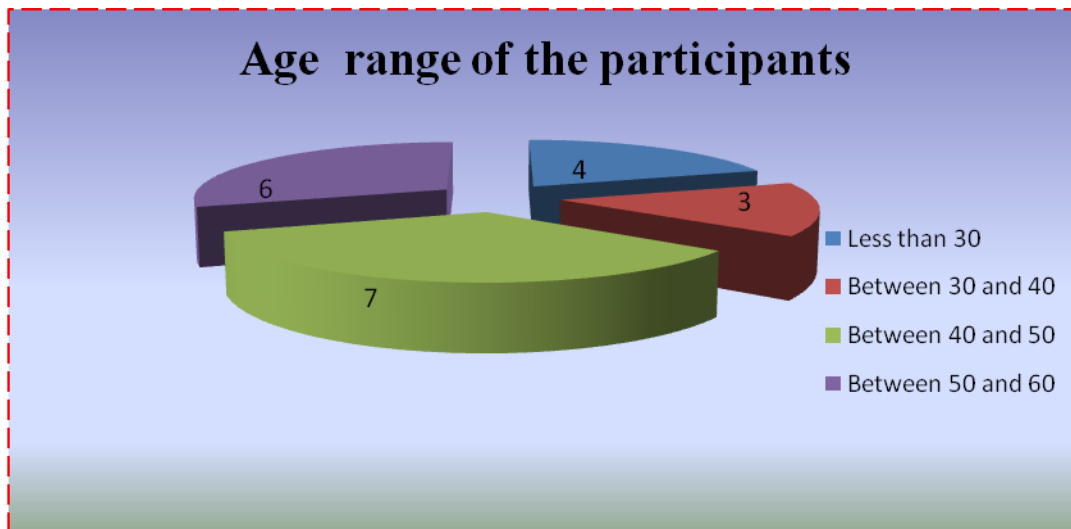
Do you think that computer services should be translated into different languages in order to ease their use?

The chart shows the answers from the participants.



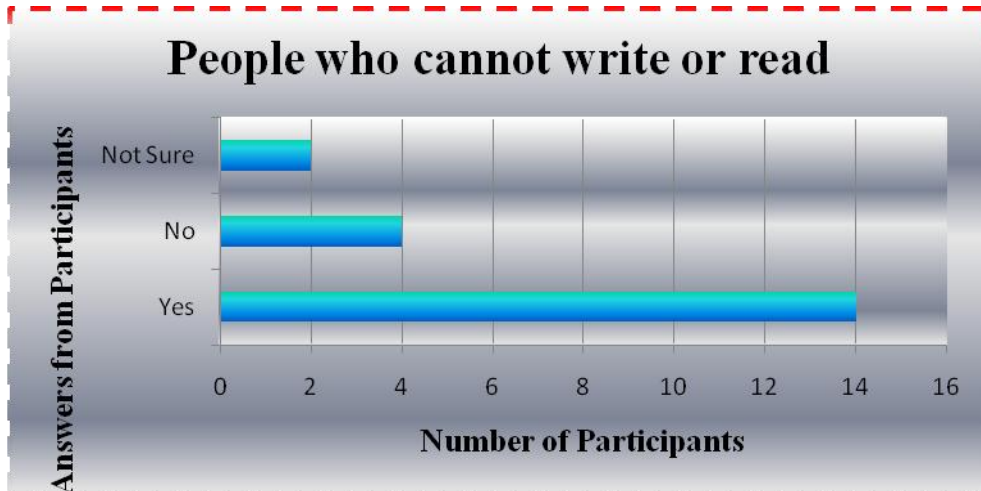
Sixteen participants answered yes; they continue by explaining that this is especially true for those who are interested and willing to learn more about how the computer works.

What is the range of your age group?



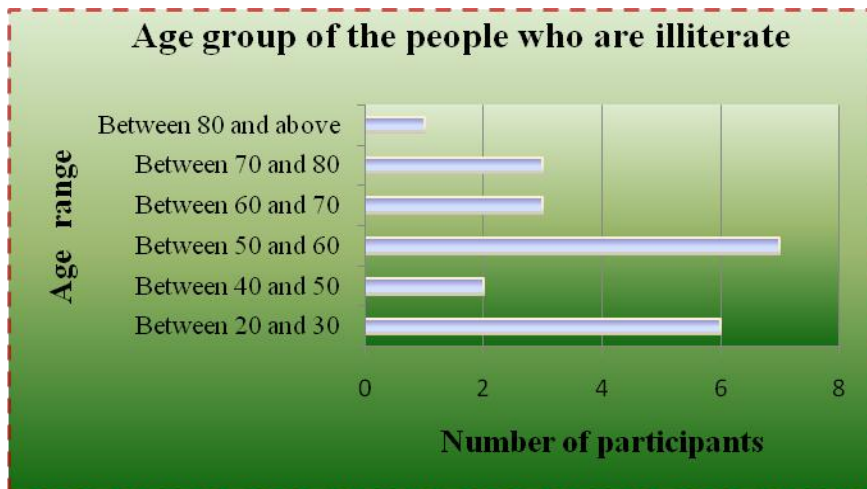
The age group of participants range between twenty to sixty years.

Are there any people in the community who cannot read or write their mother language?



The answer was yes because there are some people in the rural areas who cannot write and read isiXhosa.

If they are there in which age group?

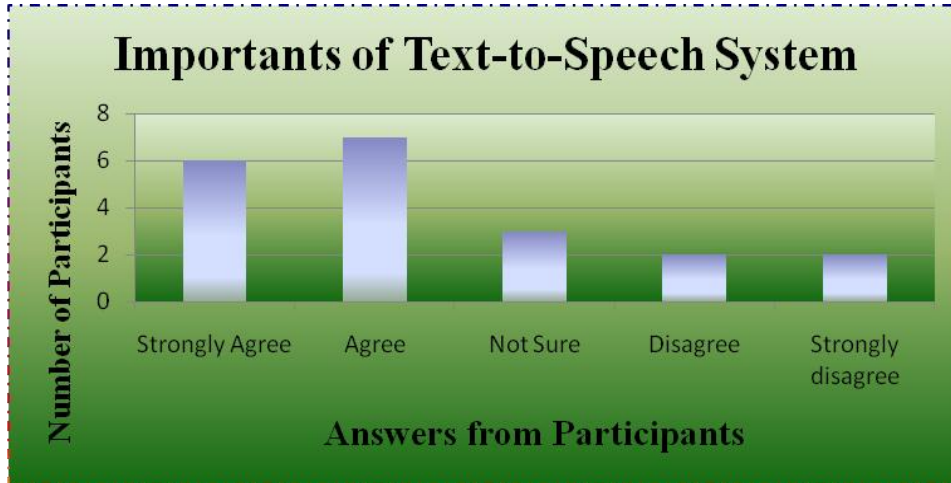


The chart shows the age group of people who cannot write and read.

12 Usability testing – Questionnaires Part 2

12.1 System usability testing

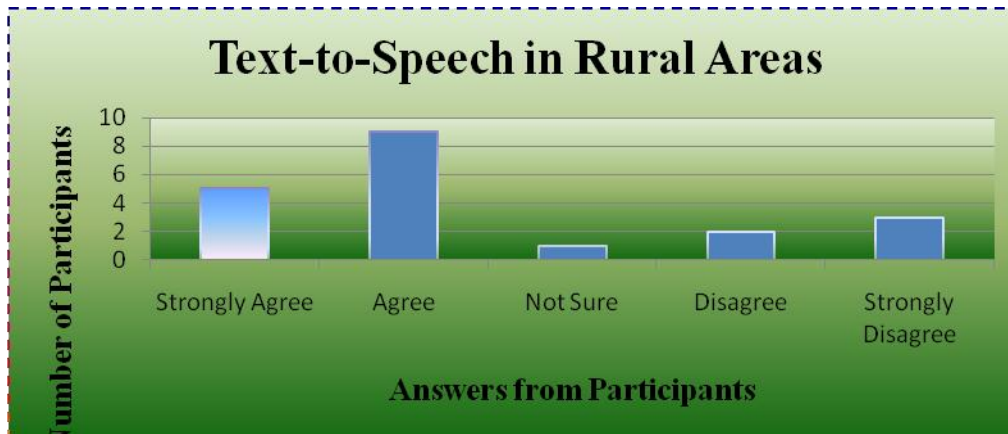
Do you think Text-To-Speech system is more important?



Do you think the system would help people who cannot write or read in the community?

All participants answered yes to this question.

Do you think the system is going to improve the use of computers in the community?



The answer was yes because they were satisfied with the system during the training.

Is the voice clear and natural sounding?

Thirteen participants said that the voice from the system was natural sounding while seven participants disagreed by saying that the system does not sound like the voice of a human being. It sounds like a

robotic system.

Does the system pronounce the clicks very well?

Twelve participants answered by saying that the system does pronounce the clicks very well while eight said that the system is not clear when it pronounces the clicks.

Is it easy to use the Text-to-Speech system?

Since the participants were given a chance to play around with the system during the test, fifteen participants answer yes while five participants said no. The five participants further explained by saying that the system included many commands that need to be considered.

Is there any need to be removed from the system?

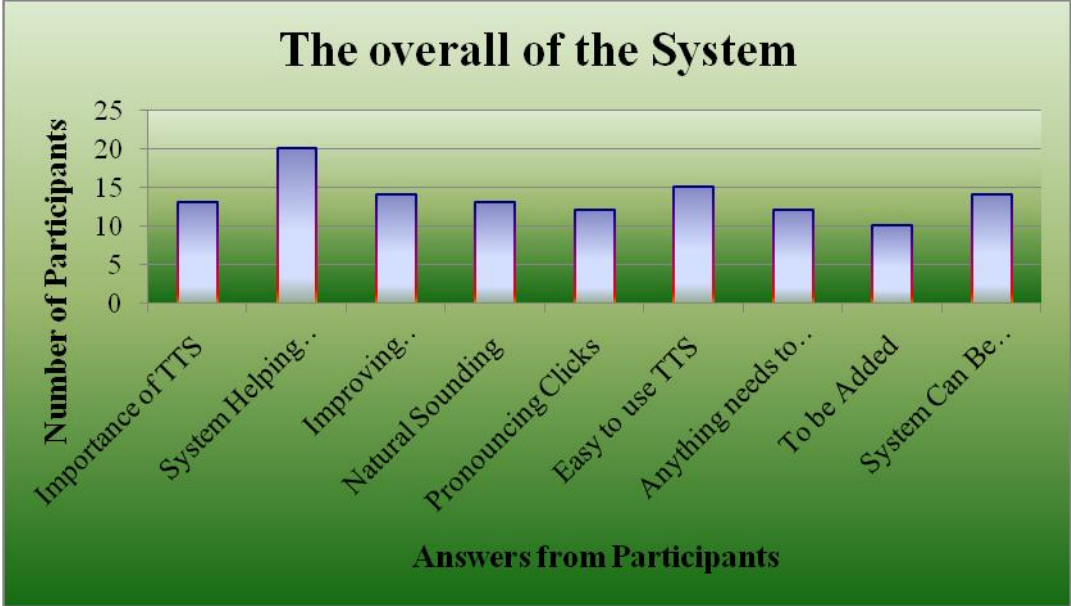
Twelve participants said no, the system is perfect there is no need to remove anything from it, while eight participants said yes. The eight participants further explained by saying that the font size in the terminal is too small and other names are not in the database.

Is there anything that needs to be added to the system?

Ten participants said yes, there is information that needs to be added, such as more words in the database so that the quality of voice can be improved. Ten participants said no, there is no need to add anything to the system because it sounds good and natural.

Can this system be deployed on the internet?

Fourteen participants said yes the system can be deployed in the internet so that it can be used throughout the country. Six participants said no because the quality of the voice is not good. The following chart shows the overall answers from participants.



The questionnaires were summarized in the above chart.